SIEMENS

SIMATIC S7

STEP 7 V5.0
Introducción y ejercicios prácticos

Guía del usuario

Este manual forma parte del paquete de documentación con la referencia:

6ES7018-4CA04-8DA0

Prólogo, Indíce

Introducción a STEP 7 — 1

Administrador SIMATIC — 2

Programación simbólica — 3

Programar el OB1 — 4

Crear un programa con FBs y DBs — 5

Configurar los módulos centrales — 6

Cargar y probar el programa — 7

Programar una función (FC) — 8

Programar un bloque de datos globales — 9

Programar una multiinstancia — 10

Configurar la periferia descentralizada — 11

Anexo A — A

Indice alfabético

03/99
C79000-G7078-C560
Versión 02

Consignas de seguridad para el usuario

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones están puestas de relieve mediante señales de precaución. Las señales que figuran a continuación representan distintos grados de peligro:

Peligro

Significa que, si no se adoptan las medidas preventivas adecuadas, **se producirá** la muerte, lesiones corporales graves o daños materiales considerables.

Precaución

Significa que, si no se adoptan las medidas preventivas adecuadas, **puede producirse** la muerte, lesiones corporales graves o daños materiales considerables.

Cuidado

Significa que, si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales o daños materiales.

Nota

Se trata de una información importante, sobre el producto o sobre una parte determinada del manual, sobre la que se desea llamar particularmente la atención.

Personal cualificado

La puesta en funcionamiento y el servicio del equipo sólo deben ser llevados a cabo conforme con este manual.

Sólo está autorizado a intervenir en este equipo el **personal cualificado**. En el sentido del manual se trata de personas que disponen de los conocimientos técnicos necesarios para poner en funcionamiento, conectar a tierra y marcar los aparatos, sistemas y circuitos de acuerdo con las normas estándar de seguridad.

Uso conforme

Considere lo siguiente:

Precaución

El equipo o los componentes del sistema sólo se podrán utilizar para los casos de aplicación previstos en el catálogo y en la descripción técnica, y sólo en unión de los equipos y componentes de proveniencia tercera recomendados y homologados por Siemens.

El funcionamiento correcto y seguro del producto presupone un transporte, un almacenamiento, una instalación y un montaje conforme a las prácticas de la buena ingeniería, así como una operación y un mantenimiento rigurosos.

Marca registrada

SIMATIC®, SIMATIC NET® y SIMATIC HMI® son marcas registradas por SIEMENS AG

Los restantes nombres y designaciones contenidos en el presente impreso pueden ser marcas registradas cuya utilización por terceros para sus fines pueden violar los derechos de los proprietarios.

Copyright © Siemens AG 1998 All rights reserved

La divulgación y reproducción de este documento, así como el uso y la comunicación de su contenido, no están autorizados, a no ser que se obtenga el consentimiento expreso para ello. Los infractores quedan obligados a la indemnización de los daños. Se reservan todos los derechos, en particular para el caso de concesión de patentes o de modelos de utilidad.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschaeftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D-90327 Nuernberg

Exención de responsabilidad

Hemos probado el contenido de esta publicación con la concordancia descrita para el hardware y el software. Sin embargo, es posible que se den algunas desviaciones que nos impiden tomar garantía completa de esta concordancia. El contenido de esta publicación está sometido a revisiones regularmente y en caso necesario se incluyen las correcciones en la siguiente edición. Agradecemos sugerencias.

© Siemens AG 1998
Sujeto a cambios sin previo aviso.

Bienvenido a STEP 7 ...

... el software estándar de SIMATIC para crear programas PLC en KOP, FUP o AWL para equipos SIMATIC S7-300 y S7-400.

Informaciones acerca de la guía de usuario

En este manual describimos las nociones básicas de STEP 7 de SIMATIC mostrando los cuadros de diálogo más importantes y los procedimientos a seguir con ayuda de ejercicios prácticos. Lo hemos estructurado de tal forma que se puede leer capítulo a capítulo en el orden deseado.

Al final de cada apartado encontrará dos cuadros enmarcados en verde: el cuadro con el fondo gris es un cuadro explicativo, mientras que el cuadro con el fondo blanco hace referencia al proceso explicado. Las instrucciones a ejecutar las hemos marcado al principio con una flecha dentro de la barra verde y al final con un punto más información adicional. Estas instrucciones pueden englobar varias páginas.

Estar acostumbrado a trabajar con el ratón, varias ventanas a la vez, menús emergentes, etc. y tener conocimientos básicos acerca de los sistemas de automatización le será de gran ayuda.

Los cursillos especiales de STEP 7 no sólo le ofrecen la posibilidad de conocer de cerca el 'Getting Started', sino también de profundizar en sus conocimientos y aprender a crear soluciones de automatización completas con STEP 7.

Requisitos para trabajar con el proyecto de ejemplo

Para realizar los ejercicios prácticos de STEP 7 incluidos en el proyecto de ejemplo se requiere:

- una unidad de programación Siemens o un PC
- el paquete de software de STEP 7 y el disquete de autorización
- un sistema de automatización SIMATIC S7-300 o S7-400
  (para el capítulo 7 "Cargar y probar el programa“)

Documentación de STEP 7

- Información básica
- Manuales de referencia

Para acceder a los manuales electrónicos una vez instalado el software STEP 7 vaya al menú **Inicio > Simatic > Manuales S7**. También se pueden pedir en cualquier sucursal de venta de Siemens. Asimismo, se puede acceder a toda la información contenida en los manuales a través de la Ayuda en pantalla de STEP7.

¡Le deseamos mucho éxito en su trabajo!

SIEMENS AG

Indice

1 Introducción a STEP 7

1.1 Resumen breve 1-1

1.2 Interacción de software y hardware 1-3

1.3 Procedimiento básico a seguir con STEP 7 1-4

1.4 Instalar STEP 7 1-5

2 Administrador SIMATIC

2.1 Arrancar el Administrador SIMATIC y crear el proyecto 2-1

2.2 Estructura del proyecto en el Administrador SIMATIC y cómo llamar a la Ayuda de STEP 7 2-4

En los capítulos 3 a 5 creará un programa sencillo.

3 Programación simbólica

3.1 Direcciones absolutas 3-1

3.2 Programación simbólica 3-2

4 Programar el OB1

4.1 Abrir la ventana KOP/AWL/FUP 4-1

4.2 Programar el OB1 en KOP 4-4

4.3 Programar el OB1 en AWL 4-8

4.4 Programar el OB1 en FUP 4-11

5 Crear un programa con FBs y DBs

5.1 Crear y abrir un bloque de función 5-1

5.2 Programar el FB1 en KOP 5-3

5.3 Programar el FB1 en AWL 5-6

5.4 Programar el FB1 en FUP 5-8

5.5 Crear bloques de datos de instancia y modificar valores actuales 5-11

5.6 Programar la llamada del bloque en KOP 5-13

5.7 Programar la llamada del bloque en AWL 5-16

5.8 Programar la llamada del bloque en FUP 5-18

En los capítulos 6 y 7 creará la configuración del hardware y probará su programa.

6 Configurar los módulos centrales

6.1 Configurar el hardware 6-1

7 Cargar y probar el programa

7.1 Establecer una conexión online 7-1

7.2 Cargar el programa en el sistema de destino 7-3

7.3 Test visualizando el estado del programa 7-6

7.4 Probar el programa con la tabla de variables 7-8

7.5 Evaluar el búfer de diagnóstico 7-12

En los capítulos 8 a 11 profundizará sus conocimientos aprendiendo nuevas funciones.

8 Programar una función (FC)

8.1 Crear y abrir la función 8-1

8.2 Programar la función 8-3

8.3 Llamada de la función en el OB1 8-6

9 Programar un bloque de datos globales

9.1 Crear y abrir el bloque de datos globales 9-1

10 Programar una multiinstancia

10.1 Crear y abrir un bloque de función de orden superior 10-1

10.2 Programar el FB10 10-2

10.3 Crear el DB10 y adaptar el valor actual 10-6

10.4 Llamada del FB10 en el OB1 10-8

11 Configurar la periferia descentralizada

11.1 Conectar la periferia descentralizada vía el bus de campo PROFIBUS-DP 11-1

Anexo A

Proyectos de ejemplo A-1

1 Introducción a STEP 7

1.1 Resumen breve

Con ayuda de ejercicios prácticos le enseñaremos lo fácil que es programar con STEP 7 en KOP, FUP o AWL.

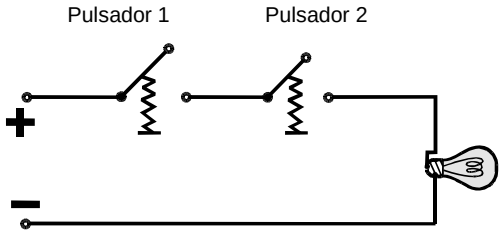
En cada capítulo encontrará instrucciones detalladas que le darán a conocer paso a paso las distintas posibilidades que ofrece STEP 7.

Crear un programa con funciones lógicas binarias

En los capítulos 2 a 7 crearemos un programa con funciones lógicas binarias. A través de las funciones lógicas programadas se accede a las entradas y salidas de la CPU (caso que existan).

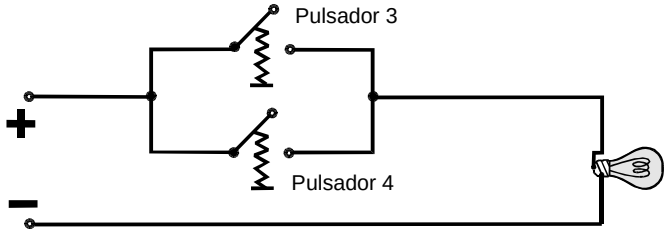
Los ejemplos de programación de la Guía del usuario se basan, entre otros, en tres funciones lógicas binarias fundamentales.

La primera función lógica binaria que programaremos más tarde es la función Y (AND). La figura siguiente ilustra esta función mediante un circuito eléctrico con dos pulsadores.



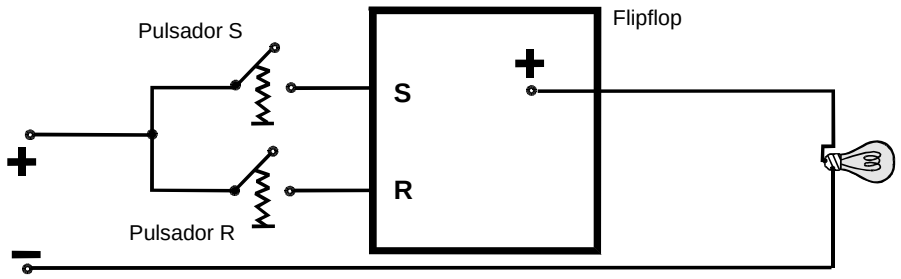
Si se oprime el pulsador 1 **y** el pulsador 2, se enciende la bombilla.

La segunda función lógica binaria es la función O (OR). Esta función también se representa mediante un circuito eléctrico con dos pulsadores.



Si se oprime el pulsador 3 **o** el pulsador 4, se enciende la bombilla.

La tercera función lógica binaria es el flipflop o función SR. Esta función reacciona en un circuito eléctrico a determinados estados de tensión y los transmite como corresponde.

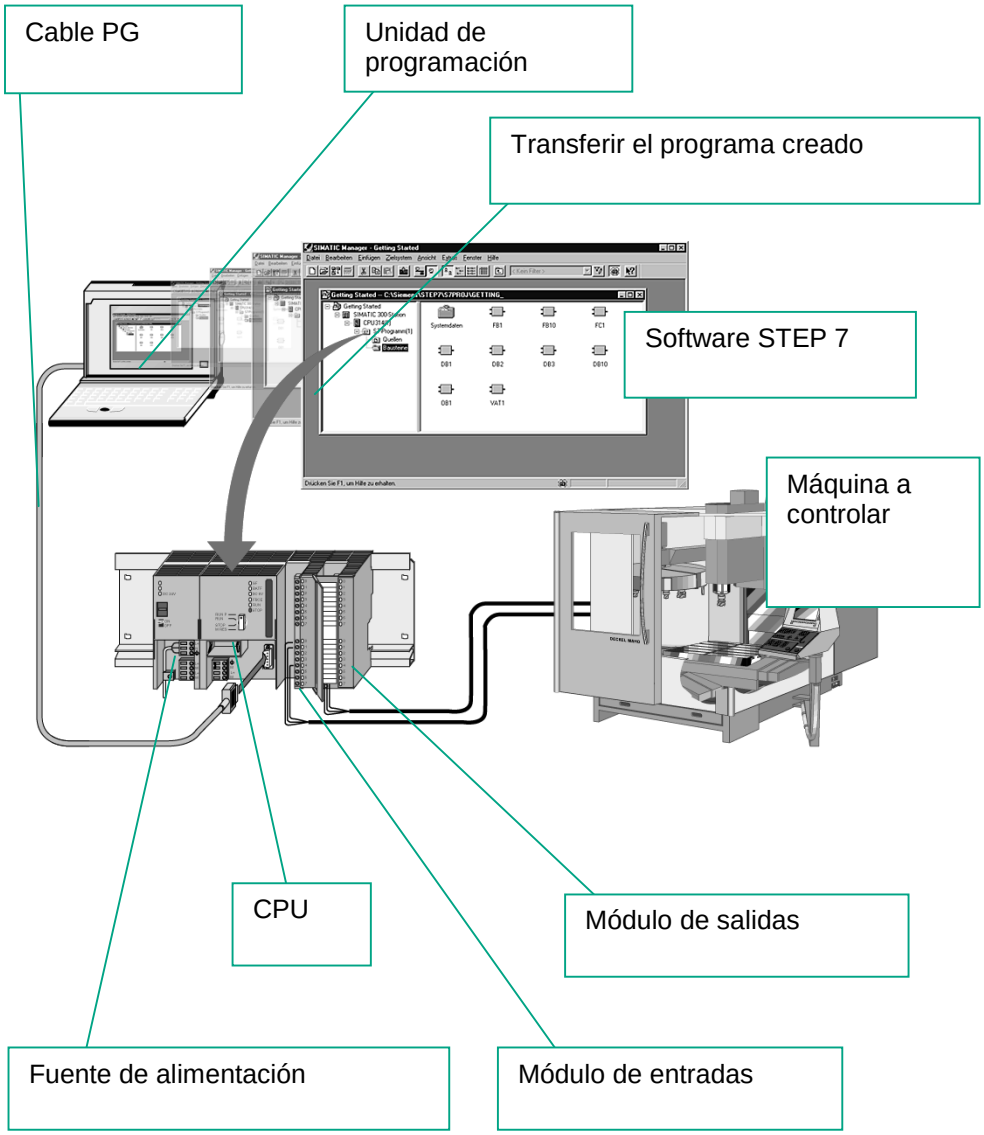


Si se oprime el pulsador S, se enciende la bombilla y permanece encendida hasta que se oprime el pulsador R.

1.2 Interacción de software y hardware

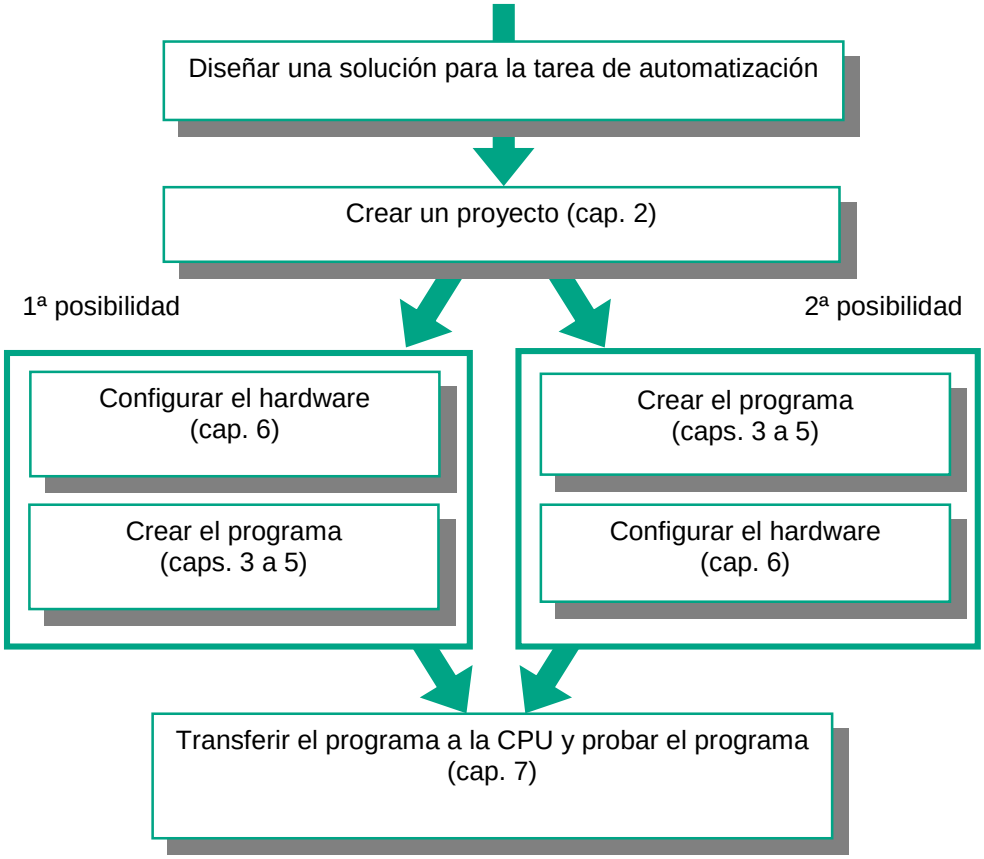
Con el software de STEP 7 crearemos un programa S7 en un proyecto. El sistema de automatización S7 se compone de una fuente de alimentación, una CPU y varios módulos de entradas/salidas (módulos E/S).

El autómata programable (PLC) vigila y controla la máquina con ayuda del programa S7. A los módulos de entradas/salidas se accede mediante direcciones.



1.3 Procedimiento básico a seguir con STEP 7

Antes de crear un proyecto debe saber que los proyectos de STEP 7 se pueden crear siguiendo el orden deseado.


Diseñar una solución para la tarea de automatización
Crear un proyecto (cap. 2)
1ª posibilidad
2ª posibilidad
Configurar el hardware (cap. 6)
Crear el programa (caps. 3 a 5)
Crear el programa (caps. 3 a 5)
Configurar el hardware (cap. 6)
Transferir el programa a la CPU y probar el programa (cap. 7)

En el caso de que el programa contenga muchas entradas y salidas recomendamos configurar primero el hardware. Así STEP 7 mostrará varias direcciones a elegir en el editor de 'HWConfig'.

Si se decide por la 2ª posibilidad, tendrá que asignar manualmente a los componentes que haya seleccionado sus direcciones respectivas, pues no podrá acceder a ellas a través de STEP 7.

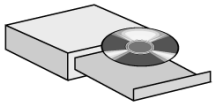
Al configurar el hardware no sólo se pueden definir direcciones, sino también modificar parámetros y propiedades de los módulos. A modo de ejemplo: en caso de utilizar varias CPUs simultáneamente, será necesario adaptar sus respectivas direcciones MPI.

Como que en el programa de ejemplo se utilizan sólo unas cuantas entradas y salidas, olvidémonos por el momento de configurar el hardware y empecemos enseguida a programar.

1.4 Instalar STEP 7

Independientemente de si desea empezar con la programación o con la configuración del hardware, primero debe instalar STEP 7. Caso que utilice una PG SIMATIC, ya tendrá el software instalado.

Al instalar el software de STEP 7 en una PG/un PC que no lo tenga preinstalado, tenga en cuenta los requisitos de software y hardware mencionados en el archivo 'Léame.wri' del CD de STEP 7 bajo **<Unidad>:\STEP 7\Disk1**



Si primero tiene que instalar STEP 7, introduzca ahora el CD. El programa de instalación arrancará automáticamente. Siga las instrucciones de instalación.

Si falla el arranque automático de la instalación, encontrará el programa 'Setup' en el CD-ROM bajo **<Unidad>:\STEP 7\Disk1\setup.exe**.

Una vez finalizada la instalación y después de reiniciar el ordenador, aparecerá el icono "Administrador SIMATIC" en su escritorio de Windows.

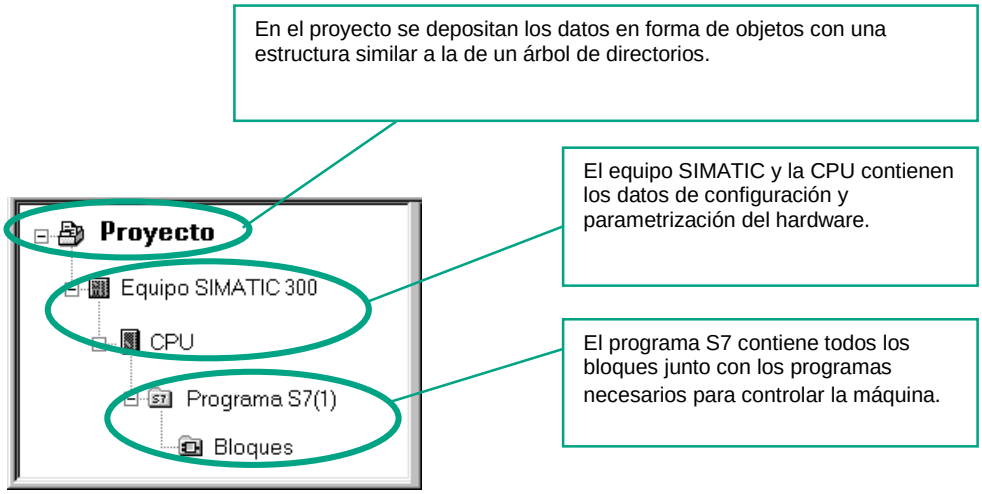
Si hace doble clic en el icono "Administrador SIMATIC" una vez finalizada la instalación, arrancará automáticamente el Asistente de STEP 7.

Para más información acerca del proceso de instalación consulte el archivo 'Léame.wri' del CD de STEP 7 bajo **<Unidad>:\STEP 7\Léame.wri**

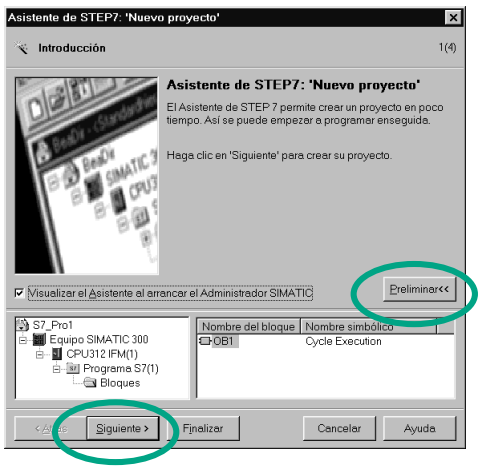
2 Administrador SIMATIC

2.1 Arrancar el Administrador SIMATIC y crear el proyecto

Al arrancar STEP 7 se abre la ventana del Administrador SIMATIC. De forma estándar, arranca asimismo el Asistente de STEP 7, el cual nos ayudará a crear el proyecto de STEP 7. Los proyectos se estructuran de tal modo que permiten depositar de forma ordenada todos los datos y programas que se necesitan durante la programación.

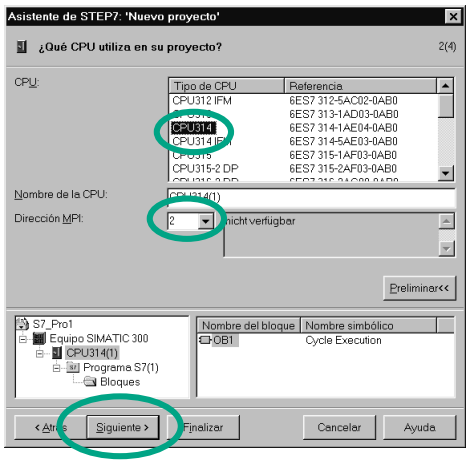


Haga doble clic en el icono **Administrador SIMATIC**. Se activará el Asistente de STEP 7.



Con el botón **Preliminar** se muestra y oculta la estructura del proyecto que se está creando.

Para que se abra el segundo cuadro de diálogo pulse el botón **Siguiente**.



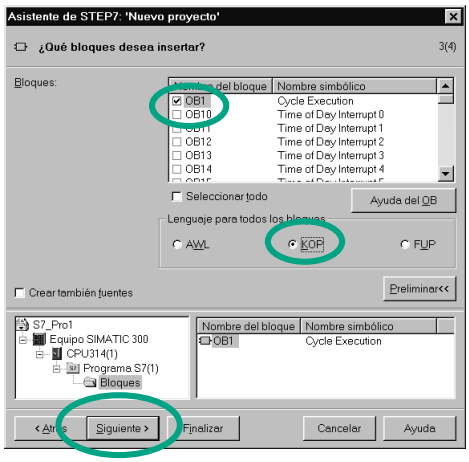
Elija la CPU 314 para el proyecto de ejemplo "GS" (Getting Started). El ejemplo se ha creado de tal modo que puede seleccionar la CPU suministrada en cualquier momento.

La dirección MPI ajustada por defecto es la 2.

Con el botón **Siguiente** se confirman los ajustes y se salta al diálogo siguiente.

Cada CPU tiene determinadas propiedades, p.ej., referentes a la capacidad de memoria o a las áreas de operandos. Por lo tanto, es preciso seleccionar la CPU antes de empezar a programar.

Para que la CPU pueda comunicarse con la PG/el PC se requiere la dirección MPI (Multi Point Interface).



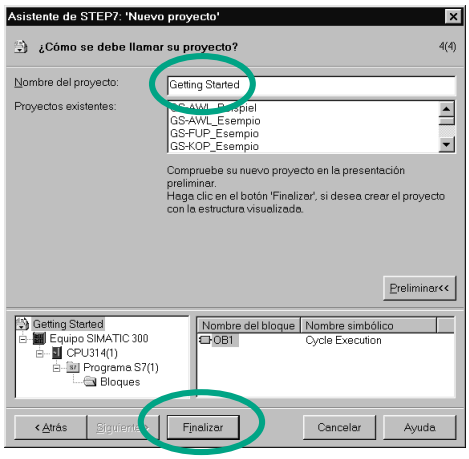
Seleccione el bloque de organización **OB1** (si aún no está seleccionado).

Elija el lenguaje de programación **KOP**, **FUP** o **AWL**.

Confirme los ajustes realizados con **Siguiente**.

El OB1 representa el nivel superior del programa y coordina los otros bloques del programa S7.

Eel lenguaje de programación se puede cambiar posteriormente.



Seleccione con un doble clic el nombre que aparece en la casilla "Nombre del proyecto" y llámelo "Getting Started".

El botón **Finalizar** permite generar el nuevo proyecto conforme a la presentación preliminar.

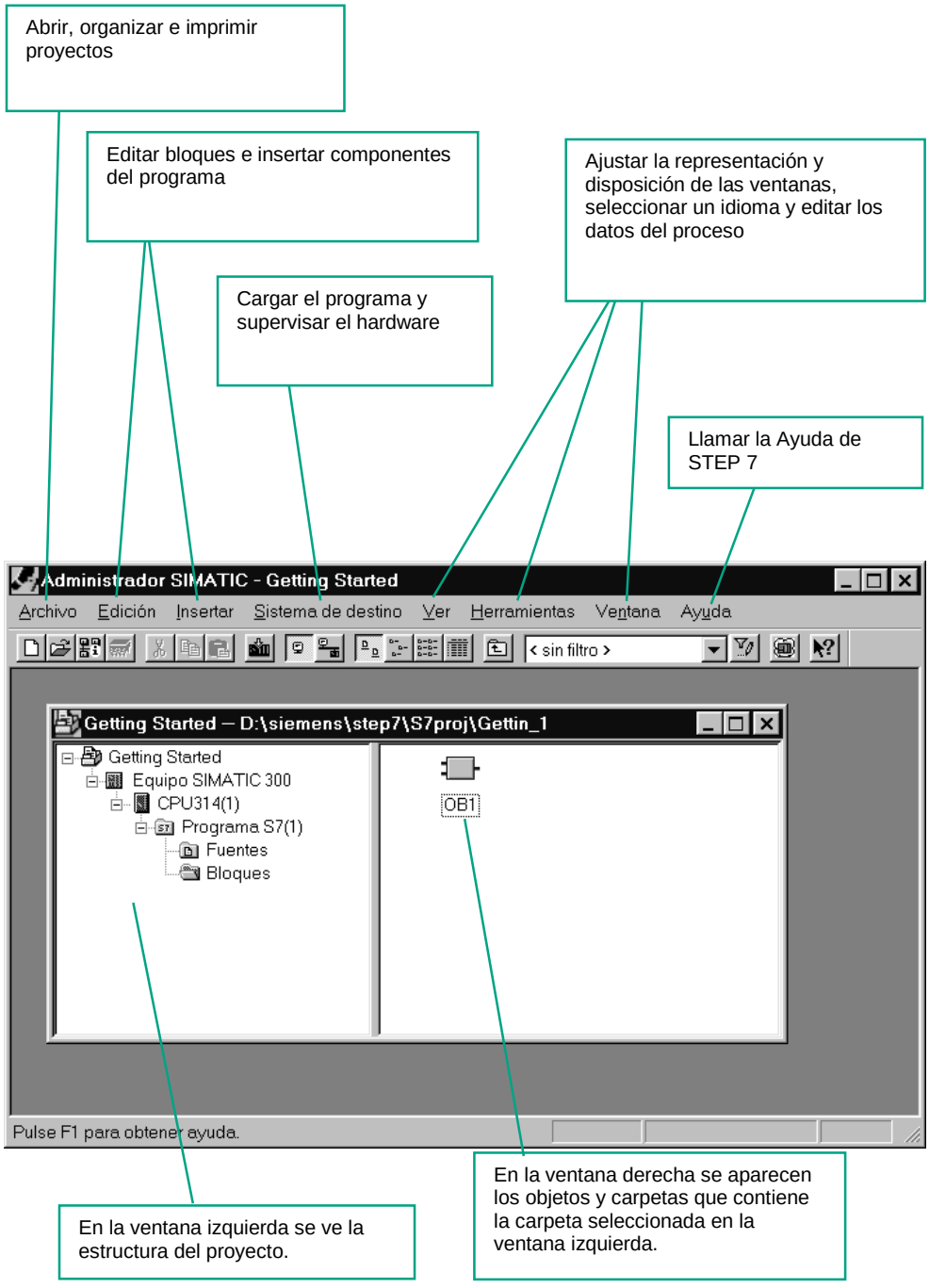
Al pulsar el botón de comando **Finalizar** se abre el Administrador SIMATIC mostrando la ventana del proyecto creado, "Getting Started". En las páginas que siguen le enseñamos lo importantes que son los archivos y las carpetas creadas y cómo trabajar con ellos de forma eficiente.

De forma estándar, el Asistente de STEP 7 se activa cada vez que se arranca el programa. Si desea modificar este ajuste, desactívelo en el primer cuadro de diálogo del Asistente. Pero no olvide que, si crea un proyecto sin ayuda del Asistente, deberá crear todos los directorios del proyecto a mano.

Para más información elija **Ayuda > Temas de Ayuda** y vaya al tema "Elaboración de proyectos".

2.2 Estructura del proyecto en el Administrador SIMATIC y cómo llamar a la Ayuda de STEP 7

Tan pronto se haya cerrado el Asistente de STEP 7 aparecerá el Administrador SIMATIC con la ventana del proyecto "Getting Started" abierta. Desde esta ventana se accede a todas las funciones y ventanas de STEP 7.

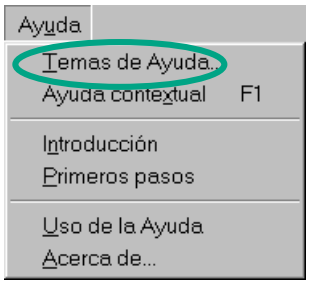

Abrir, organizar e imprimir proyectos
Editar bloques e insertar componentes del programa
Ajustar la representación y disposición de las ventanas, seleccionar un idioma y editar los datos del proceso
Cargar el programa y supervisar el hardware
Llamar la Ayuda de STEP 7
Administrador SIMATIC - Getting Started
Archivo Edición Insertar Sistema de destino Ver Herramientas Ventana Ayuda
< sin filtro >
Getting Started – D:\siemens\step7\S7proj\Gettin_1
Getting Started
Equipo SIMATIC 300
CPU314(1)
Programa S7(1)
Fuentes
Bloques
OB1
Pulse F1 para obtener ayuda.
En la ventana izquierda se ve la estructura del proyecto.
En la ventana derecha se aparecen los objetos y carpetas que contiene la carpeta seleccionada en la ventana izquierda.

Cómo llamar la Ayuda de STEP 7

F1

1ª posibilidad:

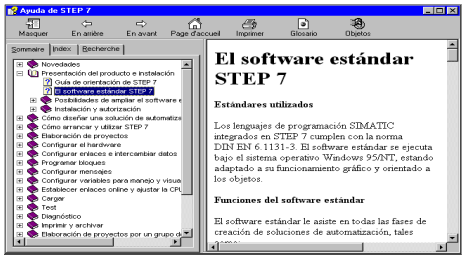
Elegir un comando de menú cualquiera y pulsar la tecla **F1**. Entonces aparecerá la Ayuda contextual del comando seleccionado.



2ª posibilidad:

Elegir el comando de menú **Ayuda > Temas de Ayuda**.

En la ventana izquierda aparece el índice con los distintos temas de Ayuda y, en la derecha, el tema elegido.



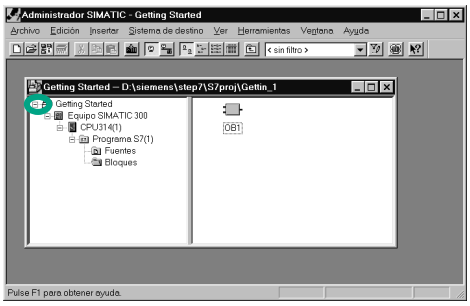
Navegue hasta el tema deseado haciendo clic en el **índice**, en el signo **+**. En la ventana derecha aparece entonces el tema seleccionado.

Con **Indice** y **Buscar** se pueden introducir términos determinados para afinar la búsqueda.

3ª posibilidad:

Hacer clic en el botón de la Ayuda y, seguidamente, en el objeto sobre el que desea obtener información.

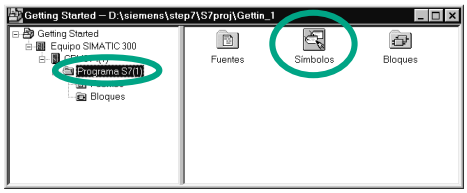
Navegar en la estructura del proyecto



Se visualiza el proyecto recién creadas con el equipo S7 seleccionado y la CPU.

Haga clic en los signos **+** ó **–** para abrir o cerrar una carpeta, respectivamente.

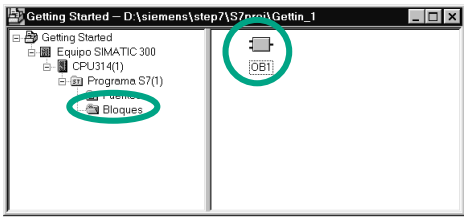
A través de los iconos visualizados en la ventana derecha puede llamar posteriormente otras funciones.



Haga clic en la carpeta **Programa S7 (1)**. Contiene los componentes necesarios del programa.

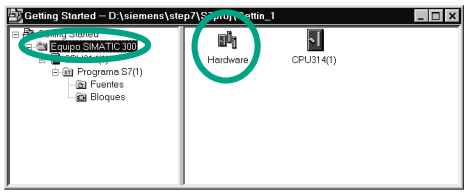
Con el icono 'Símbolos' asignaremos posteriormente nombres simbólicos a las direcciones (cap. 3).

La carpeta 'Fuentes' sirve para depositar programas fuente. En este manual no tratamos este tipo de programas.



Haga clic en la carpeta **Bloques**. Contiene el **OB1** y, más tarde, contendrá los demás bloques que vayamos programando.

A través de los bloques aprenderemos a programar en KOP, FUP y AWL (caps. 4 y 5).



Haga clic en la carpeta **Equipo SIMATIC 300**. Contiene todos los datos del proyecto que tienen que ver con el hardware.

Con el icono **Hardware** definiremos los parámetros del sistema de automatización (cap. 6).

Si necesita más software de SIMATIC (p.ej.: los paquetes opcionales PLC-SIM, el programa de simulación de hardware, o S7-GRAPH, el lenguaje gráfico de programación), también se integrarán en STEP 7. Estando en el Administrador SIMATIC se pueden abrir directamente los objetos correspondientes (p.ej.: un bloque de función S7-GRAPH).

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo diseñar una solución de automatización" y "Nociones básicas para diseñar la estructura del programa".

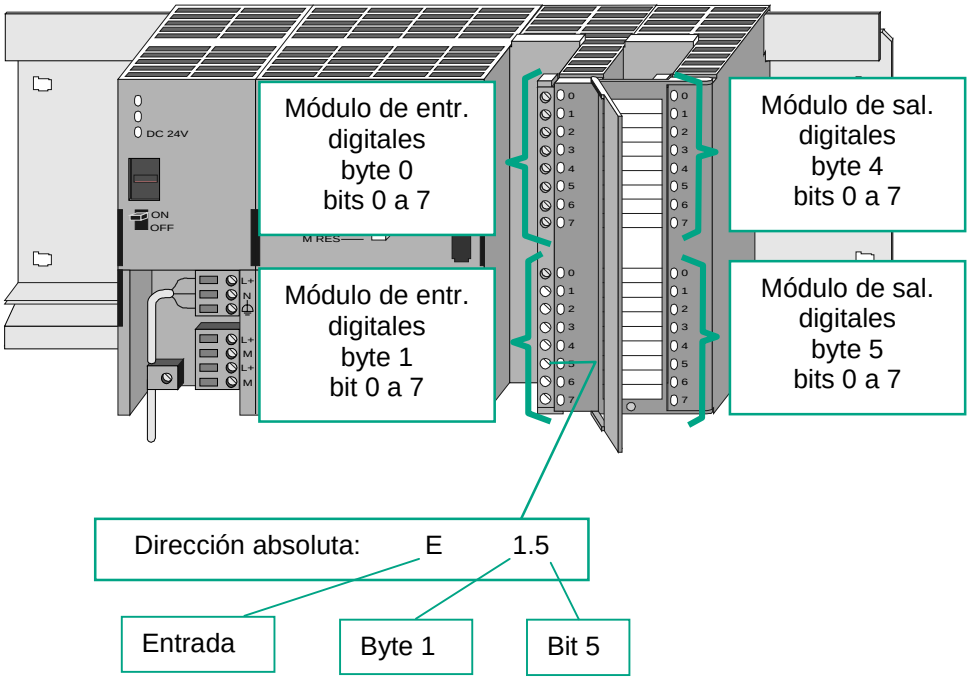
Si desea saber más acerca de los paquetes opcionales consulte el catálogo de SIMATIC "Componentes para la Integración Total en Automatización“ ST 70.

3 Programación simbólica

3.1 Direcciones absolutas

Todas las entradas y salidas tienen asignada, de forma estándar, una dirección absoluta. Esta dirección se adopta directamente en el software.

Las direcciones absolutas se pueden sustituir por nombres simbólicos definidos por el usuario.



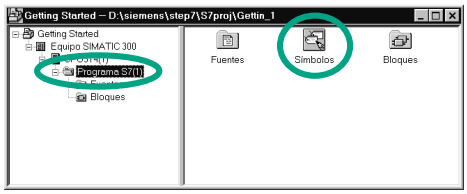
Recomendamos no hacer uso de la programación absoluta a menos que el programa S7 contenga pocas entradas y salidas.

3.2 Programación simbólica

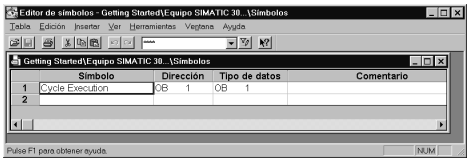
En la tabla de símbolos asigne, a todas las direcciones absolutas que utilizará más tarde en el programa un nombre simbólico y un tipo de datos; p.ej.: el símbolo 'Pulsador 1' a la entrada 'E0.1'. Estos nombres, denominados variables globales, son válidos para todos los componentes del programa.

La programación simbólica permite mejorar considerablemente la legibilidad del programa S7.

Uso del editor de símbolos



Navegue en la ventana del proyecto "Getting Started“ hasta la carpeta **Programas S7 (1)** y abra el icono **Símbolos** con un doble clic.


Editor de símbolos - Getting Started\Equipo SIMATIC 30...\Símbolos
Tabla Edición Insertar Ver Herramientas Ventana Ayuda
Getting Started\Equipo SIMATIC 30...\Símbolos
Símbolo Dirección Tipo de datos Comentario
1 Cycle Execution OB 1 OB 1
2
Pulse F1 para obtener ayuda.
NUM

Por el momento nuestra tabla de símbolos contiene sólo el bloque de organización OB1.

	Símbolo	Dirección	Tipo de datos
1	Cycle Execution	OB 1	OB 1
2			

Haga clic en el símbolo **Cycle Execution** y sobreescríbalo para nuestro ejemplo con "Programa principal“.

	Símbolo	Dirección	Tipo de datos
1	Programa principal	OB 1	OB 1
2	Lámpara verde	A 4.0	BOOL

En la línea 2 introduzca "Lámpara verde“ y "A 4.0“. El tipo de datos se añade automáticamente.

Comentario

Para introducir el comentario del símbolo haga clic en la columna 'Comentario' de la línea 1 ó 2. Al finalizar una línea pulse la tecla **Entrar** para insertar otra.

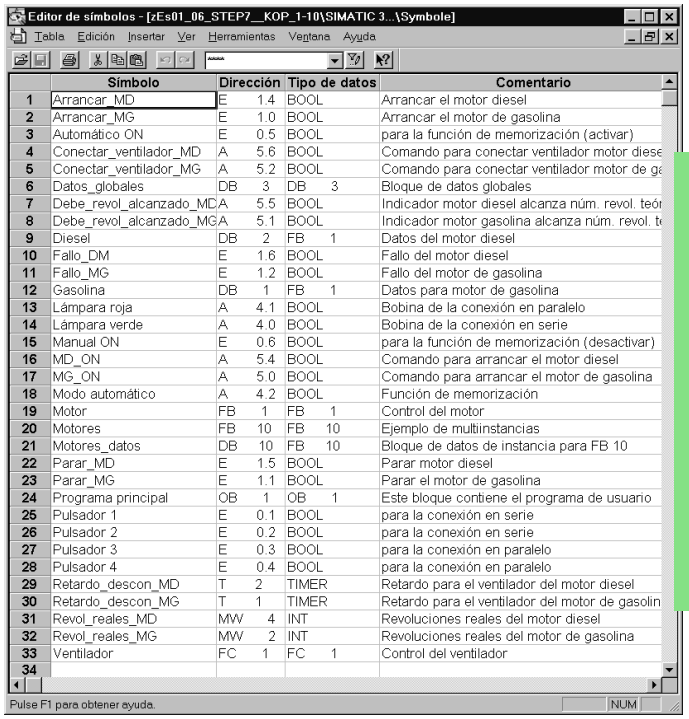
	Símbolo	Dirección	Tipo de datos
1	Programa principal	OB 1	OB 1
2	Lámpara verde	A 4.0	BOOL
3	Lámpara roja	A 4.1	BOOL

En la línea 3 introduzca "Lámpara roja" y "A 4.1“ y confirme con **Entrar**.

De este modo se asignan nombres simbólicos a las direcciones absolutas de las entradas y salidas requeridas por el programa.

Guarde los datos introducidos o modificados en la tabla de símbolos y cierre la ventana.

Como que son muchos nombres para todo el proyecto "Getting Started", recomendamos copiar la tabla de símbolos en el proyecto tal y como describimos en el siguiente capítulo.


Editor de símbolos - [zEs01_06_STEP7__KOP_1-10\SIMATIC 3...\Symbole]
Tabla Edición Insertar Ver Herramientas Ventana Ayuda
Pulse F1 para obtener ayuda.
NUM

	Símbolo	Dirección	Tipo de datos	Comentario
1	Arrancar_MD	E 1.4	BOOL	Arrancar el motor diesel
2	Arrancar_MG	E 1.0	BOOL	Arrancar el motor de gasolina
3	Automático ON	E 0.5	BOOL	para la función de memorización (activar)
4	Conectar_ventilador_MD	A 5.6	BOOL	Comando para conectar ventilador motor diese
5	Conectar_ventilador_MG	A 5.2	BOOL	Comando para conectar ventilador motor de ga
6	Datos_globales	DB 3	DB 3	Bloque de datos globales
7	Debe_revol_alcanzado_MD	A 5.5	BOOL	Indicador motor diesel alcanza núm. revol. teór
8	Debe_revol_alcanzado_MG	A 5.1	BOOL	Indicador motor gasolina alcanza núm. revol. t
9	Diesel	DB 2	FB 1	Datos del motor diesel
10	Fallo_DM	E 1.6	BOOL	Fallo del motor diesel
11	Fallo_MG	E 1.2	BOOL	Fallo del motor de gasolina
12	Gasolina	DB 1	FB 1	Datos para motor de gasolina
13	Lámpara roja	A 4.1	BOOL	Bobina de la conexión en paralelo
14	Lámpara verde	A 4.0	BOOL	Bobina de la conexión en serie
15	Manual ON	E 0.6	BOOL	para la función de memorización (desactivar)
16	MD_ON	A 5.4	BOOL	Comando para arrancar el motor diesel
17	MG_ON	A 5.0	BOOL	Comando para arrancar el motor de gasolina
18	Modo automático	A 4.2	BOOL	Función de memorización
19	Motor	FB 1	FB 1	Control del motor
20	Motores	FB 10	FB 10	Ejemplo de multiinstancias
21	Motores_datos	DB 10	FB 10	Bloque de datos de instancia para FB 10
22	Parar_MD	E 1.5	BOOL	Parar motor diesel
23	Parar_MG	E 1.1	BOOL	Parar el motor de gasolina
24	Programa principal	OB 1	OB 1	Este bloque contiene el programa de usuario
25	Pulsador 1	E 0.1	BOOL	para la conexión en serie
26	Pulsador 2	E 0.2	BOOL	para la conexión en serie
27	Pulsador 3	E 0.3	BOOL	para la conexión en paralelo
28	Pulsador 4	E 0.4	BOOL	para la conexión en paralelo
29	Retardo_descon_MD	T 2	TIMER	Retardo para el ventilador del motor diesel
30	Retardo_descon_MG	T 1	TIMER	Retardo para el ventilador del motor de gasolin
31	Revol_reales_MD	MW 4	INT	Revoluciones reales del motor diesel
32	Revol_reales_MG	MW 2	INT	Revoluciones reales del motor de gasolina
33	Ventilador	FC 1	FC 1	Control del ventilador
34				

Aquí puede ver la tabla de símbolos del programa S7 de ejemplo "Getting Started" en AWL.

Generalmente se crea una tabla de símbolos por programa S7, es decir, independientemente del lenguaje de programación que se elija.

En la tabla de símbolos se aceptan todos los caracteres imprimibles (p.ej.: diéresis, espacios en blanco etc.).

El tipo de datos que se inserta automáticamente en la tabla de símbolos define el tipo de señal a procesar para la CPU. STEP 7 utiliza, entre otros, los tipos de datos siguientes:

BOOL BYTE WORD DWORD	Combinaciones binarias desde 1 bit (tipo BOOL) hasta 32 bits (DWORD).
CHAR	Un carácter del juego de caracteres ASCII.
INT DINT REAL	Valores numéricos (p.ej.: para calcular expresiones aritméticas).
S5TIME TIME DATE TIME_OF_DAY	Valores de hora y fecha de STEP 7 (p.ej.: para ajustar la fecha o introducir la hora)

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques" y "Definir símbolos".

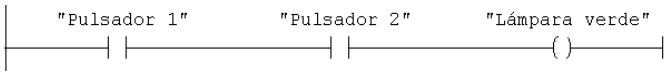
4 Programar el OB1

4.1 Abrir la ventana KOP/AWL/FUP y el OB1

Elija uno de los lenguajes KOP, AWL o FUP

STEP 7 permite crear programas S7 en los lenguajes de programación KOP, AWL o FUP. En la práctica, y también en el ejemplo de este capítulo, debe decidirse por uno de los tres lenguajes.

KOP (esquema de contactos)
apropiado, entre otros, para usuarios que provienen de la industria electrotécnica



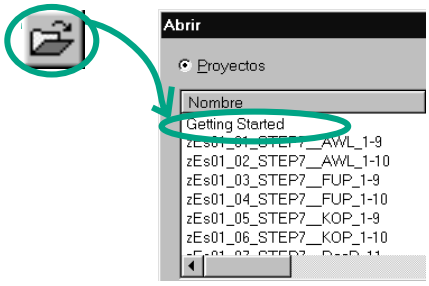
AWL (lista de instrucciones)
apropiada, entre otros, para usuarios que provienen del campo de la informática

```
U     "Pulsador 1"
U     "Pulsador 2"
=     "Lámpara verde"
```

FUP (diagrama de funciones)
apropiado, entre otros, para usuarios que estén familiarizados con la técnica de circuitos

Ahora se abrirá el bloque OB1 en el lenguaje de programación en el que haya sido creado con el Asistente del proyecto. De todos modos, el lenguaje ajustado se puede ser cambiar en todo momento.

Copiar la tabla de símbolos y abrir el OB1

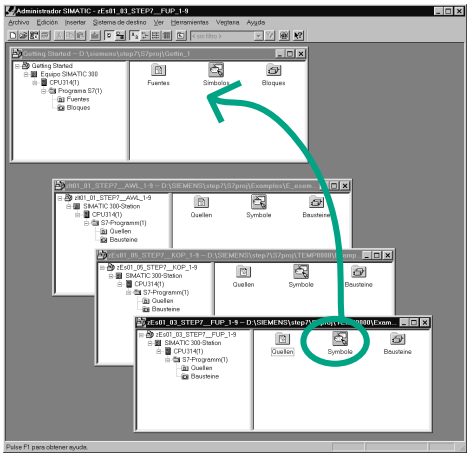


Dado el caso, abra el "Getting Started" que ha creado. Para ello, haga clic en el icono **Abrir**, eligiendo "Getting Started“ y confirme con **Aceptar**.

Dependiendo del lenguaje de programación elegido, abra también el proyecto:

- zEs01_05_STEP7__**KOP**_1-9,
- zEs01_01_STEP7__**AWL**_1-9
  o
- zEs01_03_STEP7__**FUP**_1-9

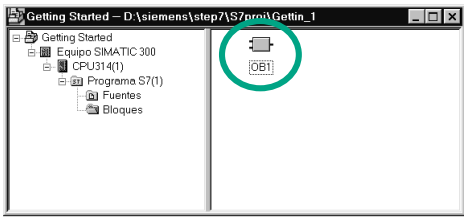
Aquí le presentamos los tres proyectos de ejemplo.



Navegue en el „zEs01_XXX“ hasta llegar a **Símbolos** y copie este icono mediante ’Drag and Drop’ (arrastrar y soltar) en la carpeta **Programa S7** de la ventana de su proyecto "Getting Started“.

A continuación cierre la ventana „zEs01_XXX“.

'Drag and Drop' consiste en hacer clic con el ratón en el objeto deseado y arrastrarlo hasta la posición deseada manteniendo pulsada la tecla del ratón. Al soltarla, el objeto se deposita allí donde se encuentre en ese momento.



Haga doble clic en el **OB1** del proyecto "Getting Started". Así se abrirá la ventana KOP/AWL/FUP del proyecto.

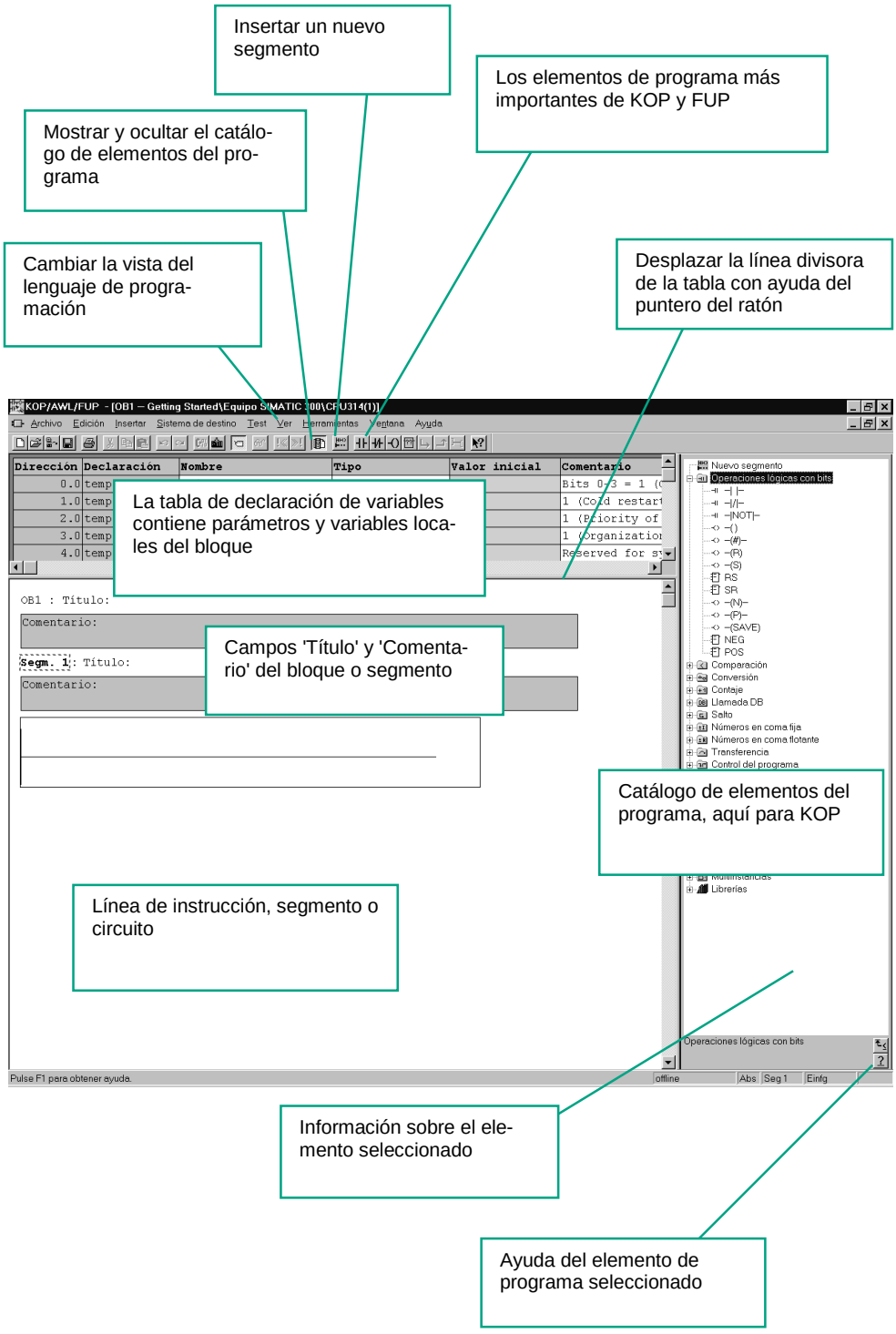
En STEP 7 el OB1 es procesado cíclicamente por la CPU. Esta lee el programa contenido en el bloque línea por línea y ejecuta los comandos. Cuando la CPU vuelve a encontrarse en la primera línea del programa, significa que ha finalizado un ciclo. El tiempo transcurrido hasta entonces se denomina 'tiempo de ciclo'.

Dependiendo del lenguaje de programación que haya elegido consulte el apt. 4.2 (programación con KOP), el apt. 4.3 (programación con AWL) o el apt. 4.4 (programación con FUP).

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques" o "Crear bloques y librerías"

La ventana KOP/AWL/FUP

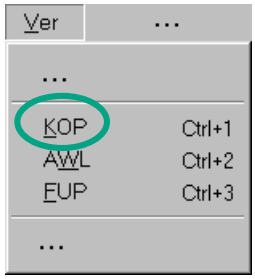
En la ventana KOP/AWL/FUP se programan todos los bloques. Como ejemplo para los tres lenguajes le mostramos aquí la ventana KOP.


Insertar un nuevo segmento
Los elementos de programa más importantes de KOP y FUP
Mostrar y ocultar el catálogo de elementos del programa
Cambiar la vista del lenguaje de programación
Desplazar la línea divisora de la tabla con ayuda del puntero del ratón
La tabla de declaración de variables contiene parámetros y variables locales del bloque
Campos 'Título' y 'Comentario' del bloque o segmento
Catálogo de elementos del programa, aquí para KOP
Línea de instrucción, segmento o circuito
Información sobre el elemento seleccionado
Ayuda del elemento de programa seleccionado

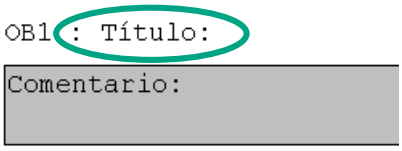
4.2 Programar el OB1 en KOP

A continuación vamos a programar una conexión en serie, una conexión en paralelo y la operación 'Flip-flop de activación/desactivación' en KOP (esquema de contactos).

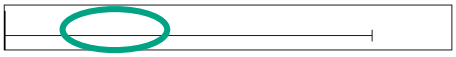
Programar una conexión en serie en KOP



Si no está ajustado el lenguaje de programación KOP, ajústelo eligiendo el comando de menú **Ver > KOP**.



Haga clic en el área **Título** del OB1 e introduzca, p.ej., "Programa principal. Se ejecuta cíclicamente".



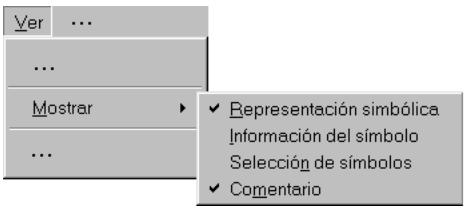
Seleccione un circuito para el primer elemento.

Haga clic en el botón que representa un contacto normalmente abierto en la barra de herramientas e insértelo.

Repita el procedimiento e inserte un segundo contacto abierto.

Inserte una bobina al final del circuito.

Para completar la conexión en serie sólo falta asignar las direcciones a los contactos normalmente abiertos y a la bobina.

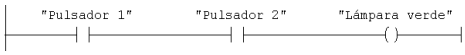


Compruebe si está activada la representación simbólica.

Haga clic en **??.?** e introduzca el nombre simbólico "Pulsador 1" (no olvide las comillas).
Confirme con **Entrar**.

Para el segundo contacto normalmente abierto introduzca el nombre simbólico "Pulsador 2".

Introduzca el nombre "Lámpara verde" para la bobina.



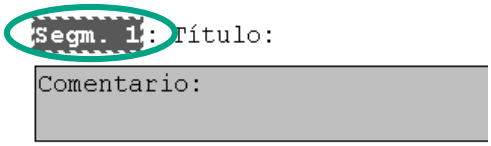
La conexión en serie está programada.

Si ya no hay más símbolos marcados en rojo, guarde el bloque.

Los símbolos se marcan de color rojo cuando no aparecen en la tabla de símbolos o bien cuando se detecta un error sintáctico.

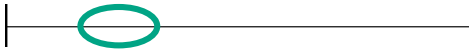
Los nombres simbólicos también se pueden insertar directamente desde la tabla de símbolos. Para ello haga clic en **??.?** y elija **Insertar > Símbolo**. Navegue por la lista desplegable hasta el nombre deseado y selecciónelo. Así se aceptará automáticamente el nombre simbólico seleccionado.

Programar una conexión en paralelo en KOP



Seleccione el **segmento 1**.

Inserte otro segmento.

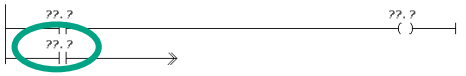


Vuelva a seleccionar el circuito.

Inserte un contacto normalmente abierto y una bobina.

Seleccione la barra vertical del circuito.

Inserte una rama paralela.



Inserte otro contacto normalmente abierto en la rama paralela.

Cierre la rama (si es necesario, seleccione la flecha inferior).

Para completar la conexión en paralelo sólo falta asignar las direcciones pertinentes.

Para asignar las direcciones simbólicas proceda como lo hemos hecho para la conexión en serie.

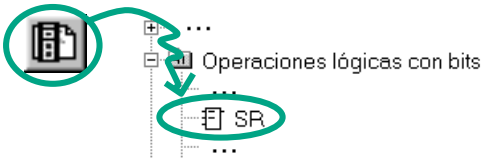
Sobreescriba el contacto abierto superior con "Pulsador 3", el inferior con "Pulsador 4" y la bobina con "Lámpara roja".

Guarde el bloque.

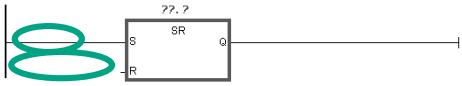
Programar una función de memorización en KOP

Seleccione el segmento 2 e inserte otro segmento.

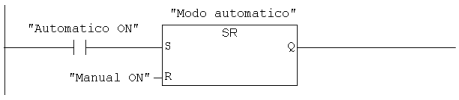
Vuelva a seleccionar el circuito.



Navegue por el catálogo de elementos del programa pasando por **Operaciones lógicas con bits** hasta llegar al elemento **SR** y haga doble clic para insertarlo.



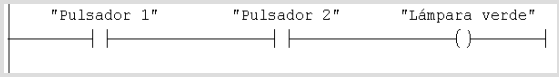
Inserte un contacto normalmente abierto antes de la entrada S y otro antes de la entrada R.



Introduzca el siguiente nombre simbólico para el elemento SR:
contacto superior "Automático ON“,
contacto inferior "Manual ON“,
elemento SR "Modo automático“.

Guarde el bloque y cierre la ventana.

Si desea ver la diferencia entre el direccionamiento absoluto y el simbólico, desactive el comando de menú **Ver > Mostrar > Representación simbólica**.



Ejemplo:
direccionamiento simbólico en KOP

Ejemplo:
direccionamiento absoluto en KOP

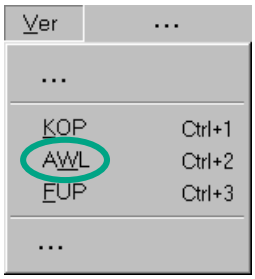
El salto de línea del direccionamiento simbólico se modifica en la ventana KOP/AWL/FUP del programa con el comando **Herramientas > Preferencias > KOP/FUP > Campo del operando (ancho)**.
Puede ajustarlo entre el 10° y el 24° carácter.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Editar instrucciones KOP en el área de instrucciones".

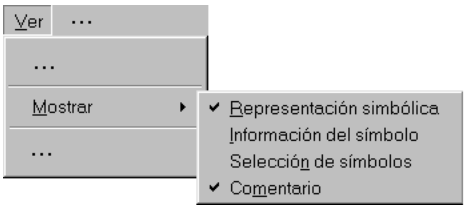
4.3 Programar el OB1 en AWL

A continuación programaremos una instrucción Y (AND), una instrucción O (OR) y las instrucciones de memorización Activar y Desactivar en AWL (lista de instrucciones).

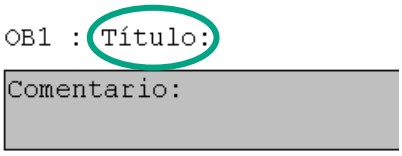
Programar una instrucción U (Y) en AWL



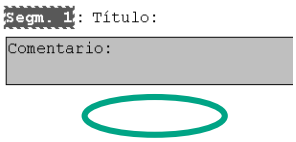
Si no está ajustado el lenguaje de programación AWL, ajústelo eligiendo el comando de menú **Ver > AWL**.



Compruebe si está activada la representación simbólica.



Haga clic en el campo **Título** del OB1 e introduzca, por ejemplo, "Programa principal. Se ejecuta cíclicamente".



Seleccione el área destinada a la primera instrucción.

```
U     "Pulsador 1"
```

Escriba en la primera línea del programa una U (Y) seguida de un espacio en blanco y el símbolo "Pulsador 1" (entre comillas).

Cierre la línea pulsando **Entrar**. Entonces el cursor saltará a la línea siguiente.

```
U     "Pulsador 1"
U     "Pulsador 2"
=     "Lámpara verde"
```

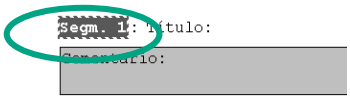
Complete la instrucción U (Y) tal y como mostramos a la izquierda.

La instrucción U está programada. Si ya no hay más botones marcados en rojo, guarde el bloque.

Los símbolos se marcan de color rojo cuando no aparecen en la tabla de símbolos o bien cuando se detecta un error sintáctico.

Los nombres simbólicos también se pueden insertar directamente desde la tabla de símbolos. Para ello haga clic en **??.?** y elija **Insertar > Símbolo**. Navegue por la lista desplegable hasta el nombre deseado y selecciónelo. Así se aceptará automáticamente el nombre simbólico seleccionado.

Programar una instrucción O en AWL



Seleccione el **segmento 1**.

Inserte otro segmento y vuelva a seleccionar el área de entrada.

```
O     "Pulsador 3"
```

Introduzca una O y el símbolo "Pulsador 3" (igual que hemos hecho con la instrucción U).

```
O     "Pulsador 3"
O     "Pulsador 4"
=     "Lámpara roja"
```

Acabe de programar la instrucción O y guárdela.

Programar una instrucción de memorización en AWL

Seleccione el segmento 2 e inserte otro segmento.

```
U     "Automatico ON"
```

Escriba en la primera línea la instrucción U (Y) y el nombre simbólico "Automático ON".

```
U     "Automatico ON"
S     "Modo automatico"
U     "Manual ON"
R     "Modo automatico"
```

Acabe de programar la instrucción de memorización y guárdela. Cierre el bloque.

Si desea ver la diferencia entre el direccionamiento absoluto y el simbólico, desactive el comando de menú **Ver > Mostrar > Representación simbólica**.

```
U     "Pulsador 1"
U     "Pulsador 2"
=     "Lámpara verde"
```

Ejemplo:
direccionamiento simbólico en AWL

```
U     E     0.1
U     E     0.2
=     A     4.0
```

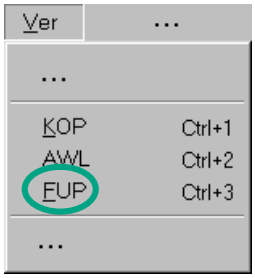
Ejemplo:
direccionamiento absoluto en AWL

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Editar instrucciones AWL en el área de instrucciones".

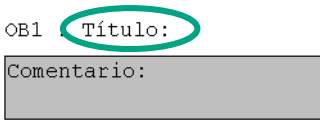
4.4 Programar el OB1 en FUP

A continuación programaremos una función U (Y), una función O y una función de memorización en FUP (diagrama de funciones).

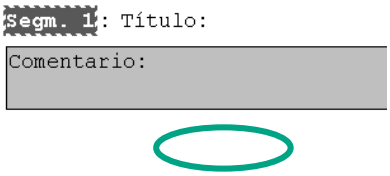
Programar una función U (Y) en FUP



Si no está ajustado el lenguaje de programacion FUP, ajústelo eligiendo el comando de menú **Ver > FUP**.

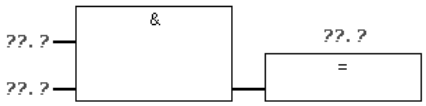


Haga clic en el campo **Título** del OB1 e introduzca, por ejemplo, "Programa principal. Se ejecuta cíclicamente".

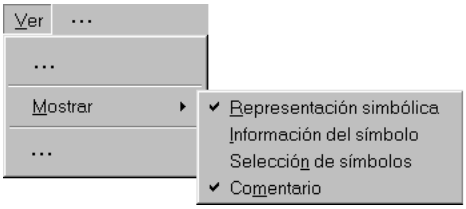


Seleccione el área prevista para la función U (Y) (se encuentra bajo el campo del comentario).

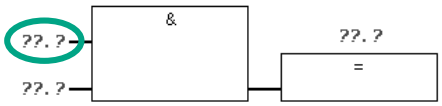
Inserte un cuadro Y (**&**) y una asignación (**=**).



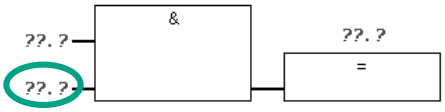
Para completar la función U (Y) sólo falta asignar una dirección a los distintos elementos.



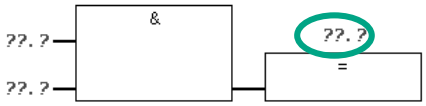
Compruebe si está activada la representación simbólica.



Haga clic en **??.?** e introduzca el nombre simbólico "Pulsador 1" (no olvide las comillas).
Confirme pulsando **Entrar**.



Introduzca el nombre simbólico "Pulsador 2" para la segunda entrada.



Introduzca el nombre "Lámpara verde" para la asignación.

La función Y está programada.

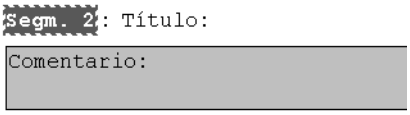
Si ya no hay más botones marcados en rojo, guarde la instrucción.

Los símbolos se marcan de color rojo cuando no aparecen en la tabla de símbolos o bien cuando se detecta un error sintáctico.

Los nombres simbólicos también se pueden insertar directamente desde la tabla de símbolos. Para ello haga clic en **??.?** y elija **Insertar > Símbolo**. Navegue por la lista desplegable hasta el nombre deseado y selecciónelo. Así se aceptará automáticamente el nombre simbólico seleccionado.

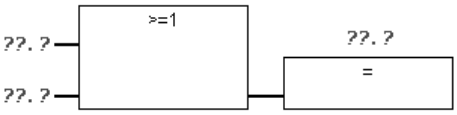
Programar una función O en FUP

Inserte otro segmento.

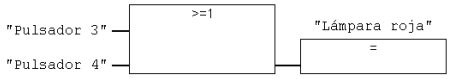


Vuelva a seleccionar el campo de entrada para la función O (OR).

Inserte un cuadro O (≥**1**) y una asignación (**=**).



Para completar la función O sólo falta asignar una dirección a los distintos elementos. Proceda como lo hemos hecho con la función U (Y).

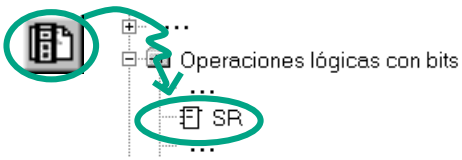


Introduzca
"Pulsador 3" para la entrada superior,
"Pulsador 4" para la entrada inferior y
"Lámpara roja" para la asignación.

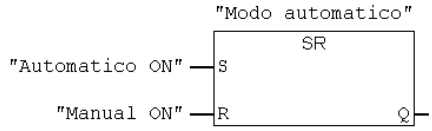
Guarde el bloque.

Programar una función de memorización en FUP

Seleccione el segmento 2 e inserte otro segmento. Vuelva a seleccionar el área de entrada (se encuentra bajo el campo del comentario).



Navegue por el catálogo de los elementos del programa pasando por **Operaciones lógicas con bits** hasta llegar al elemento **SR** y haga doble clic en el mismo para insertarlo.

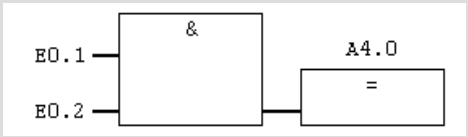


Introduzca los siguientes nombres simbólicos
para el elemento SR:
activar "Automático ON",
desactivar "Manual ON" y la marca "Modo automático".

Guarde el bloque y cierre la ventana.

Si desea ver la diferencia entre el direccionamiento absoluto y el simbólico, desactive el comando de menú **Ver > Mostrar > Representación simbólica**.

Ejemplo:
direccionamiento simbólico en FUP



Ejemplo:
direccionamiento absoluto en FUP

El salto de línea del direccionamiento simbólico se modifica en la ventana KOP/AWL/FUP del programa con el comando **Herramientas > Preferencias > KOP/FUP > Campo del operando (ancho)**. Puede ajustarlo entre el 10° y el 24° carácter.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Editar instrucciones FUP en el área de instrucciones"

5 Crear un programa con FBs y DBs

5.1 Crear y abrir un bloque de función

El bloque de función (FB) está subordinado al bloque de organización y contiene una parte del programa a la que se puede acceder siempre que se desee desde el OB1. Los parámetros formales y los datos estáticos del bloque de función se guardan por separado en un bloque de datos DB asignado al FB.

Ahora programaremos el bloque de función (FB1, nombre simbólico "Motor", v. tabla de símbolos, pág. 3-3) en la ventana KOP/AWL/FUP. Para ello utilizaremos el mismo lenguaje que en el capítulo 4 ("Programar el OB1").

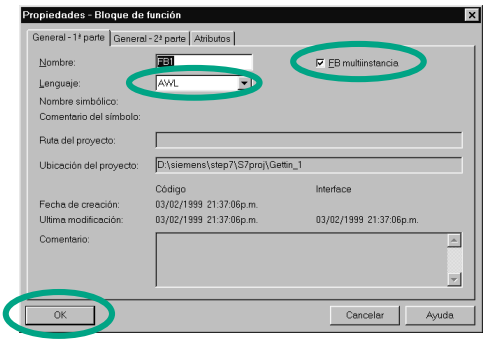
Tiene que haber copiado la tabla de símbolos en el proyecto "Getting Started". Si aún no lo ha hecho, lea primero la pág. 4-2, "Copiar la tabla de símbolos", y luego prosiga con este capítulo.

Si es necesario, abra el proyecto "Getting Started".

Navegue hasta la carpeta **Bloques** y ábrala.

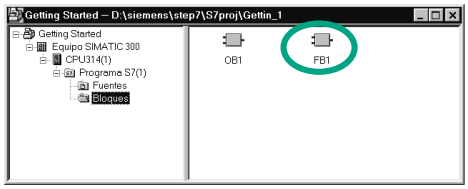
Haga clic con la tecla derecha del ratón en la mitad derecha de la ventana.

El menú contextual que aparece al pulsar la tecla derecha del ratón contiene los comandos más utilizados de la barra de menús. Inserte el objeto **Bloque de función**.



Haciendo doble clic en el FB1 se abrirá la ventana KOP/AWL/FUP.

En el cuadro de diálogo "Propiedades – Bloque de función“ seleccione el lenguaje en el que desea programar el bloque, active la opción **FB multiinstancia** y pulse **Aceptar** para que se guarden los ajustes realizados.



El bloque de función **FB1** se insertará en la carpeta 'Bloques'.

Dependiendo del lenguaje de programación que haya elegido siga leyendo en el apartado 5.2 (KOP), en el apartado 5.3 (AWL) o en el apartado 5.4 (FUP).

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques" y "Crear bloques y librerías"

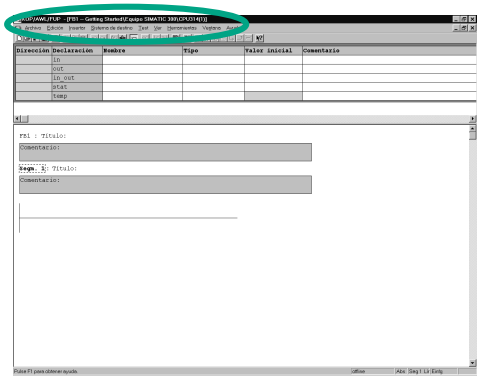
5.2 Programar el FB1 en KOP

Ahora le enseñaremos cómo programar un bloque de función que controla y supervisa, a modo de ejemplo, un motor de gasolina y un motor diesel, a saber: con un bloque de datos distinto para cada motor.

Las señales "específicas del motor" son transferidas por el bloque de organización al bloque de función en calidad de parámetros del bloque, por lo que deben figurar en la tabla de declaración de variables como parámetros de entrada y salida (declaración "in“ y "out“).

Para ello es importante que sepa programar conexiones en serie, conexiones en paralelo y funciones de memorización en STEP 7.

Primero rellenaremos la tabla de declaración de variables



La ventana KOP/AWL/FUP está abierta y el comando **Ver > KOP** (lenguaje de programación) activado.

Fíjese en el encabezado; verá que aparece el FB1, ya que ha abierto la ventana del programa haciendo doble clic en el mismo.

Introduzca las declaraciones siguientes en la tabla de declaración de variables.

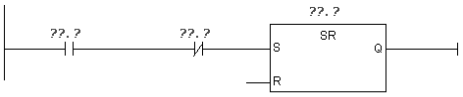
Para ello haga clic en una de las celdas y escriba el nombre y el comentario que ve en la figura.

Seleccione el tipo de datos del menú contextual **Datos simples** que aparece al pulsar la tecla derecha del ratón. Pulse **Entrar**; así el cursor saltará a la columna siguiente y se insertará otra línea.

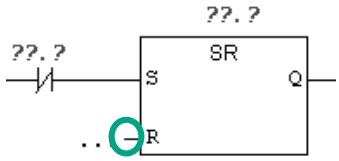
Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
0.0	in	Einschalten	BOOL	FALSE	Conectar motor
0.1	in	Ausschalten	BOOL	FALSE	Desconectar motor
0.2	in	Stoerung	BOOL	FALSE	Fallo del motor; conduce a una desconexión
2.0	in	Drehzahl_Ist	INT	0	revoluciones reales del motor
4.0	out	Motor_Ein	BOOL	FALSE	Conexión del motor
4.1	out	Soll_Dreh_erreicht	BOOL	FALSE	Número de revoluciones alcanzado
	in_out				
6.0	stat	Drehzahl_Soll	INT	1500	Número de revoluciones solicitado
	temp				

Los nombres de los parámetros del bloque que figuran en la tabla de declaración sólo pueden contener letras, cifras y el carácter de subrayado.

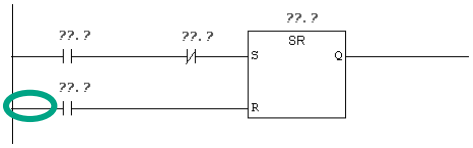
Programar el arranque y paro de un motor



Inserte un contacto normalmente abierto, uno cerrado y un elemento SR en serie en el segmento 1 encima de los símbolos correspondientes o extrayéndolos del catálogo de elementos del programa.

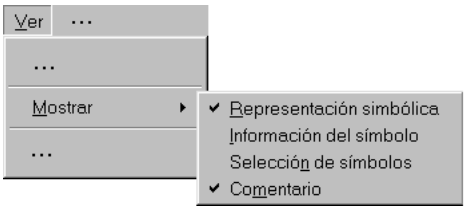


A continuación seleccione la parte del circuito inmediatamente anterior a la entrada R.



Inserte otro contacto normalmente abierto y seleccione la parte del circuito inmediatamente anterior al contacto.

Inserte un contacto normalmente cerrado paralelamente al contacto abierto.

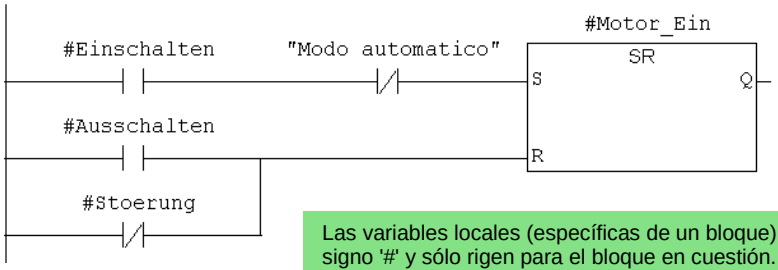


Compruebe si está activada la representación simbólica.

Seleccione los signos de interrogación e introduzca los nombres tal y como figuran en la tabla de declaración (el signo '#' se introduce automáticamente).

Introduzca el nombre simbólico "Modo automático" para el contacto normalmente cerrado de la conexión en serie.

A continuación guarde el programa.

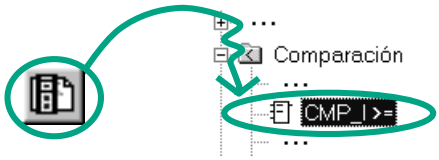


Las variables locales (específicas de un bloque) se marcan con el signo '#' y sólo rigen para el bloque en cuestión.

Las variables globales se indican entre comillas, se definen en la tabla de símbolos y son válidas para todo el programa.

El estado de señal "Modo_automático" es definido en el OB1 (segmento 3, v. pág. 4-7) por otro elemento SR y se consulta en el FB1.

Programar una supervisión para las revoluciones



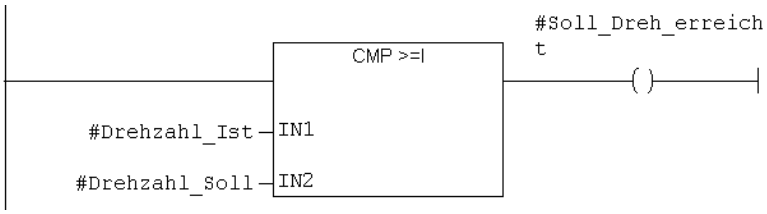
Inserte otro segmento y seleccione el circuito.

A continuación navegue por el catálogo de elementos del programa hasta llegar al **comparador** e inserte un **CMP>=1**.

Inserte además una bobina en el mismo circuito.

Vuelva a seleccionar los signos de interrogación e introduzca los nombres de la bobina y el comparador tal y como figuran en la tabla de declaración de variables.

A continuación guarde el programa.



¿Cuándo se arranca y cuándo se para el motor?

Cuando el estado de señal de la variable #Arrancar es "1“ y el de la variable "Modo automático“ es "0“, arranca el motor. Para ello es imprescindible que se niegue (contacto normalmente cerrado) la variable "Modo automático“.

Cuando el estado de señal de la variable #Parar es "1“ o cuando el de la variable #Fallo es "0“, se para el motor. Para ello es imprescindible que se niegue la variable #Fallo (#Fallo es una señal "activa a cero“; es decir, normalmente está a "1“, sólo en el caso de que se produzca un fallo se pone a "0“).

¿Cómo supervisa el comparador las revoluciones del motor?

El comparador compara la variable #Drehzahl_Ist (#Revol_reales) con la variable #Drehzahl_Soll (#Revol teóricas) y arroja el resultado por medio de la variable #Soll_Dreh_erreicht (#Revol_alcanzado) (el estado de señal es "1").

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Editar la tabla de declaración de variables" o "Editar instrucciones KOP en el área de instrucciones".

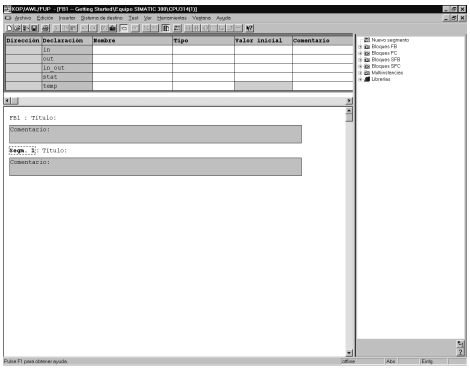
5.3 Programar el FB1 en AWL

Ahora le enseñaremos cómo programar un bloque de función que controla y supervisa, a modo de ejemplo, un motor de gasolina y un motor diesel, a saber: con un bloque de datos distinto para cada motor.

Las señales "específicas del motor" son transferidas por el bloque de organización al bloque de función en calidad de parámetros del bloque, por lo que deben figurar en la tabla de declaración de variables como parámetros de entrada y salida (declaración "in" y "out").

Para ello es importante que sepa programar conexiones en serie, conexiones en paralelo y funciones de memorización en STEP 7.

Primero rellenaremos la tabla de declaración de variables



La ventana KOP/AWL/FUP está abierta y el comando **Ver > AWL** (lenguaje de programación) activado.

Fíjese en el encabezado; verá que aparece el FB1, ya que ha abierto la ventana del programa haciendo doble clic en el mismo.

Introduzca las declaraciones siguientes en la tabla de declaración de variables.

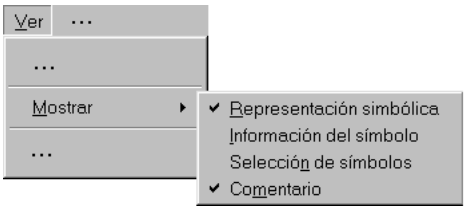
Para ello haga clic en una de las celdas y escriba el nombre y el comentario que ve en la figura.

Seleccione el tipo del menú contextual **Datos simples** que aparece al pulsar la tecla derecha del ratón. Pulse **Entrar**; así el cursor saltará a la columna siguiente y se insertará otra línea.

Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
0.0	in	Einschalten	BOOL	FALSE	Conectar motor
0.1	in	Ausschalten	BOOL	FALSE	Desconectar motor
0.2	in	Stoerung	BOOL	FALSE	Fallo del motor; conduce a una desconexión
2.0	in	Drehzahl_Ist	INT	0	revoluciones reales del motor
4.0	out	Motor_Ein	BOOL	FALSE	Conexión del motor
4.1	out	Soll_Dreh_erreicht	BOOL	FALSE	Número de revoluciones alcanzado
	in_out				
6.0	stat	Drehzahl_Soll	INT	1500	Número de revoluciones solicitado
	temp				

Los nombres de los parámetros del bloque de la tabla de declaración sólo pueden contener letras, cifras y el carácter de subrayado.

Programar el arranque y paro de un motor



Compruebe si está activada la representación simbólica.

```
U     #Einschalten
UN    "Modo automático"
S     #Motor_Ein
O     #Ausschalten
ON    #Stoerung
R     #Motor_Ein
```

Introduzca las instrucciones que le mostramos a la izquierda en el segmento 1.

Las variables locales (específicas de un bloque) se marcan con el signo '#' y sólo son válidas para el bloque en cuestión.

Las variables globales se indican entre comillas, se definen en la tabla de símbolos y rigen para todo el programa.

El estado de señal "Modo automático" es definido en el OB1 (segmento 3, v. pág. 4-10) por otro elemento SR y se consulta ahora en el FB1.

Programar una supervisión para las revoluciones

```
L     #Drehzahl_Ist
L     #Drehzahl_Soll
>=I
=     #Soll_Dreh_erreicht
```

Inserte otro segmento e introduzca las instrucciones que le mostramos a la izquierda. A continuación cierre el programa.

¿Cuándo se arranca y cuándo se para el motor?

Cuando el estado de señal de la variable #Arrancar es "1" y el de la variable "Modo automático" es "0", arranca el motor. Para ello es imprescindible que se niegue (contacto normalmente cerrado) la variable "Modo_automático".

Cuando el estado de señal de la variable #Parar es "1" o cuando el de la variable #Fallo es "0", se para el motor. Para ello es imprescindible que se niegue la variable #Fallo (#Fallo es una señal "activa a cero"; es decir, normalmente está a "1", sólo en el caso de que se produzca un fallo se pone a "0").

¿Cómo supervisa el comparador las revoluciones del motor?

El comparador compara la variable #Drehzahl_Ist (#Revol_reales) con la variable #Drehzahl_Soll (#Revol teóricas) y arroja el resultado por medio de la variable #Soll_Dreh_erreicht (#Revol_alcanzado) (el estado de señal es "1").

Para amás información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Editar la tabla de declaración de variables" o "Editar instrucciones AWL en el área de instrucciones".

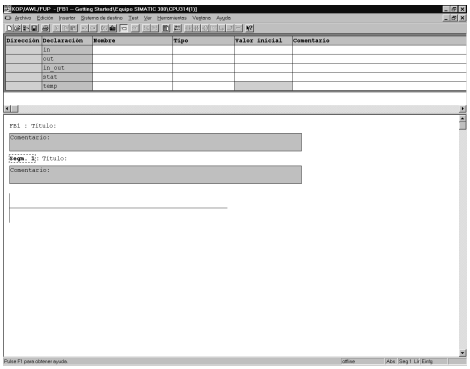
5.4 Programar el FB1 en FUP

Ahora le enseñaremos cómo programar un bloque de función que controla y supervisa, a modo de ejemplo, un motor de gasolina y un motor diesel, a saber: con un bloque de datos distinto para cada motor.

Las señales "específicas del motor" son transferidas por el bloque de organización al bloque de función en calidad de parámetros del bloque, por lo que deben figurar en la tabla de declaración de variables como parámetros de entrada y salida (declaración "in" y "out").

Para ello es importante que sepa programar conexiones en serie, conexiones en paralelo y funciones de memorización en STEP 7.

Primero rellenaremos la tabla de declaración de variables



La ventana KOP/AWL/FUP está abierta y el comando **Ver > FUP** (lenguaje de programación) activado.

Fíjese en el encabezado; verá que aparece el FB1, ya que ha abierto la ventana del programa haciendo doble clic en el mismo.

Introduzca las declaraciones siguientes en la tabla de declaración de variables.

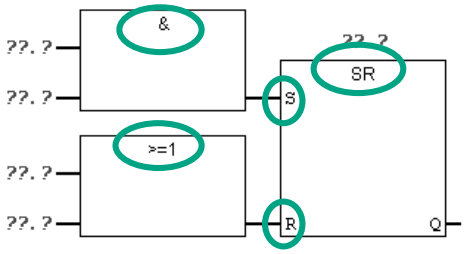
Para ello haga clic en una de las celdas y escriba el nombre y el comentario que ve en la figura.

Seleccione el tipo del menú contextual **Datos simples** que aparece al pulsar la tecla derecha del ratón. Pulse **Entrar**; así el cursor saltará a la columna siguiente y se insertará otra línea.

Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
0.0	in	Einschalten	BOOL	FALSE	Conectar motor
0.1	in	Ausschalten	BOOL	FALSE	Desconectar motor
0.2	in	Stoerung	BOOL	FALSE	Fallo del motor; conduce a una desconexión
2.0	in	Drehzahl_Ist	INT	0	revoluciones reales del motor
4.0	out	Motor_Ein	BOOL	FALSE	Conexión del motor
4.1	out	Soll_Dreh_erreicht	BOOL	FALSE	Número de revoluciones alcanzado
	in_out				
6.0	stat	Drehzahl_Soll	INT	1500	Número de revoluciones solicitado
	temp				

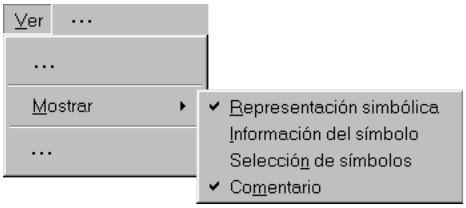
Los nombres de los parámetros del bloque de la tabla de declaración sólo pueden contener letras, cifras y el guión inferior.

Programar el arranque y paro de un motor



Inserte una función SR (directorio "Operaciones lógicas con bits") en el segmento 1 extrayéndola del catálogo de los elementos del programa.

A continuación inserte un cuadro Y para la entrada S (activar) y un cuadro O en la entrada R (desactivar).



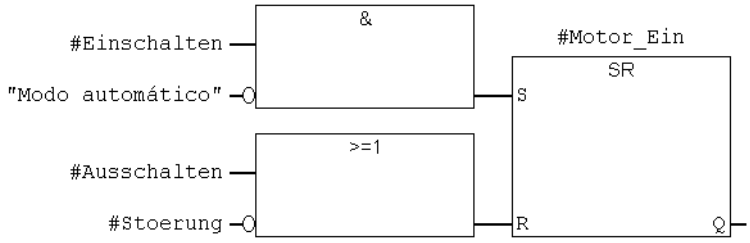
Compruebe si está activada la representación simbólica.

Haga clic en **??.?** e introduzca los nombres tal y como figuran en la tabla de declaración (el signo '#' se introduce automáticamente).

Asegúrese de que se direccione una entrada de la función Y con el nombre simbólico "Modo automático".

Niegue las entradas "Modo_automático" y #Fallo con los símbolos correspondientes (botones de la barra de herramientas).

A continuación guarde el programa.

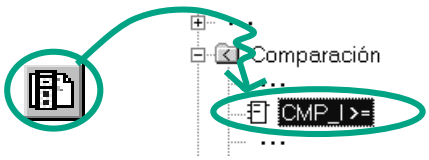


Las variables específicas de un bloque se marcan con el signo '#' y sólo rigen para el bloque en cuestión.

Las variables globales rigen entre comillas, se definen en la tabla de símbolos y son válidas para todo el programa.

El estado de señal "Modo automático" es definido en el OB1 (segmento 3, v. pág. 4-14) por otro elemento SR y se consulta en el FB1.

Programar una supervisión para las revoluciones

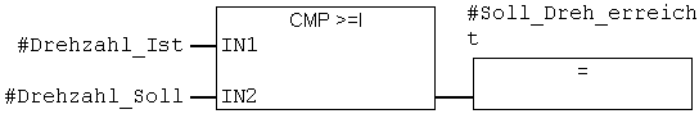


Inserte otro segmento y seleccione el área de entrada.

A continuación navegue por el catálogo de elementos del programa hasta llegar al **comparador** e inserte un **CMP>=1**.

Inserte una asignación de salida detrás del comparador y direccione las entradas con los nombres de la tabla de declaración de variables.

A continuación guarde el programa.



¿Cuándo se arranca y cuándo se para el motor?

Cuando el estado de señal de la variable #Arrancar es "1" y el de la variable "Modo automático" es "0", se pone en marcha el motor. Para ello es imprescindible que se niegue (contacto cerrado) la variable "Modo automático".

Cuando el estado de señal de la variable #Parar es "1" o cuando el de la variable #Fallo es "0", se para el motor. Para ello es imprescindible que se niegue la variable #Fallo (#Fallo es una señal "activa a cero"; es decir, normalmente está a "1", sólo en el caso de que se produzca un fallo se pone a "0").

¿Cómo supervisa el comparador las revoluciones del motor?

El comparador compara la variable #Drehzahl_Ist (#Revol_reales) con la variable #Drehzahl_Soll (#Revol teóricas) y arroja el resultado por medio de la variable #Soll_Dreh_erreicht (#Revol_alcanzado) (el estado de señal es "1").

Para amás información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Editar la tabla de declaración de variables" o "Editar instrucciones AWL en el área de instrucciones".

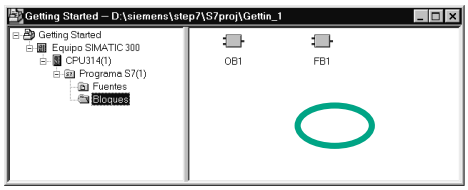
5.5 Crear bloques de datos de instancia y modificar valores actuales

Hemos programado el bloque de función FB1 ("Motor") definiendo, entre otros, parámetros específicos del motor en la tabla de declaración de variables.

Para poder programar más tarde la llamada (CALL) del OB1 al FB, debemos crear el bloque de datos correspondiente. Generalmente se asigna a cada FB un bloque de datos de instancia (DB).

El FB controla y supervisa un motor de gasolina y un motor diesel. El número teórico de revoluciones del motor de gasolina se registra en un DB y el del motor diesel en otro, por lo que tendremos que modificar el valor actual en cada caso (#N°_teór_rev).

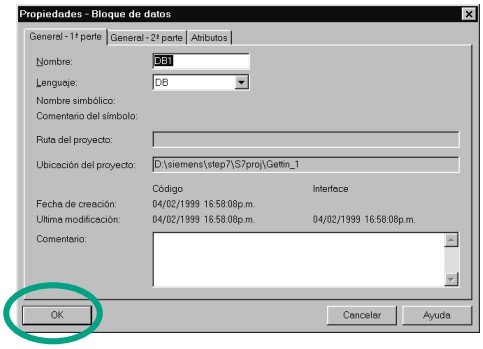
Nos podemos ahorrar un tiempo valioso si programamos un único bloque de función para ambos DB.



El proyecto "Getting Started" está abierto en el Administrador SIMATIC.

Navegue dentro del proyecto hasta llegar a la carpeta **Bloques** y haga clic con la tecla derecha del ratón en la mitad derecha de la ventana.

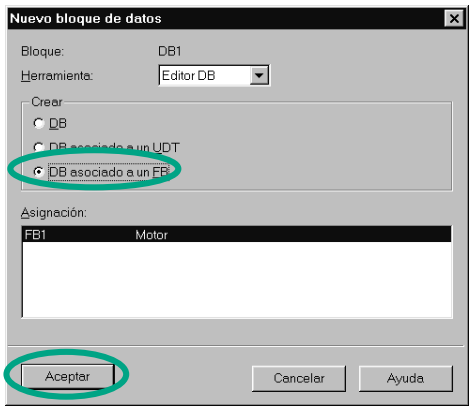
Inserte un **bloque de datos** con ayuda del menú contextual que aparece al pulsar la tecla derecha del ratón.



Confirme todos los ajustes del cuadro de diálogo "Propiedades" con **Aceptar**.

Así se insertará el bloque de datos DB1 en el proyecto "Getting Started".

Abra el **DB1** con un doble clic.

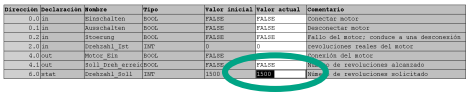


Active la opción **DB asociado a un FB** del cuadro de diálogo "Nuevo bloque de datos".

Confirme la asignación "FB1, Motor" con **Aceptar**.

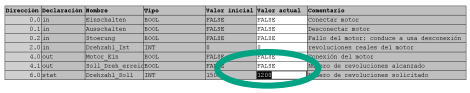
Así se abrirá la ventana KOP/AWL/FUP del programa con los datos de la tabla de declaración de variables del FB1.

Ahora debemos introducir en el DB1 los datos específicos del motor de gasolina. Pero primero elegiremos el comando de menú **Ver > Datos**.



Introduzca, para el motor de gasolina, el valor "1500" en la columna 'Valor actual' (de la línea "N°_teór_rev"). Así definirá el número máximo de revoluciones para este motor.

Guarde el DB1 y cierre la ventana del programa.



Cree el bloque de datos DB2 del mismo modo que hemos creado el DB1.

Para el motor diesel introduzca el valor "1200" como valor actual.

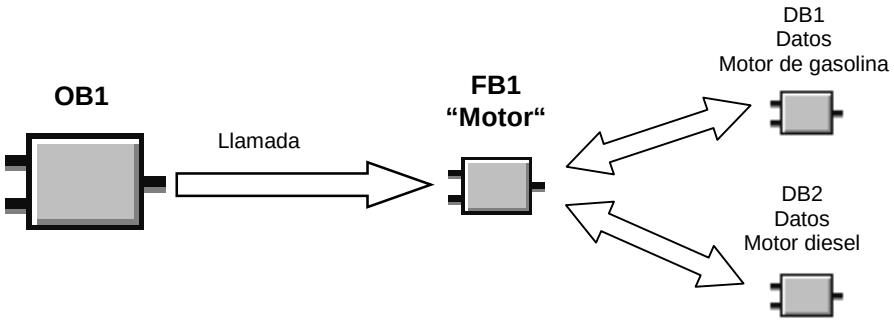
Una vez modificados los valores actuales podemos controlar ambos motores con un solo bloque de función. Para controlar más motores con el mismo bloque sólo tendríamos que crear los correspondientes bloques de datos.

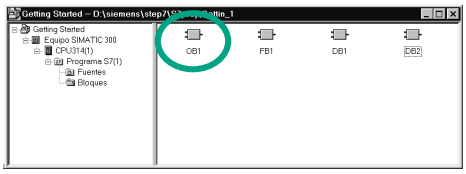
Para programar la llamada del FB en el OB1 consulte el apt. 5.6 (KOP), el apt. 5.7 (AWL) o el apt. 5.8 (FUP), dependiendo del lenguaje que haya utilizado al programar el bloque.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques" y "Crear bloques de datos".

5.6 Programar la llamada del bloque en KOP

La programación realizada en un FB no sirve de nada si este FB no es llamado por el OB1. Cada vez que se llama al FB se utiliza un bloque de datos distinto, controlándose así ambos motores.

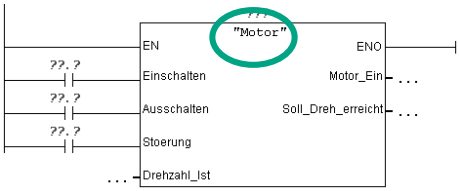




El proyecto "Getting Started" está abierto en el Administrador SIMATIC.

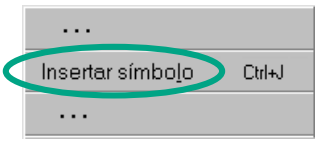
Navegue hasta la carpeta **Bloques** y abra el **OB1**.

Inserte el segmento 4 en la ventana KOP/AWL/FUP. A continuación navegue por el catálogo de elementos del programa hasta el **FB1** e insértelo.

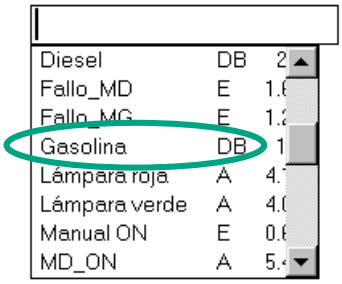


Inserte un contacto normalmente abierto antes de 'Arrancar', 'Parar' y 'Fallo'.

Haga clic en **???** encima de "Motor", y seguidamente con la tecla derecha del ratón en el cuadro de entrada.

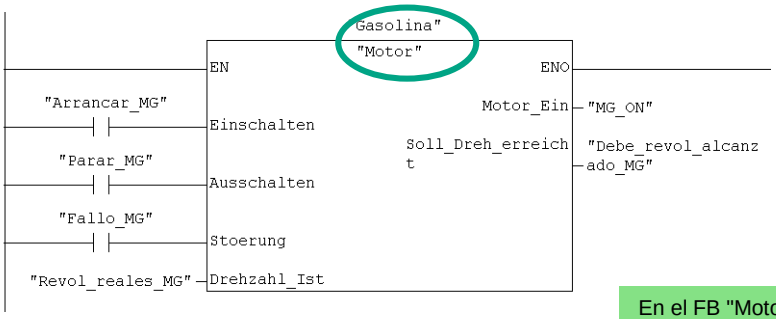


Haga clic en el menú contextual que aparece al pulsar la tecla derecha del ratón en **Insertar símbolo**. Entonces se abrirá una lista desplegable. La primera vez tardará un poco más en abrirse.



Haga clic en el bloque de datos **Gasolina**. Así aparecerá escrito entre comillas en el cuadro de entrada.

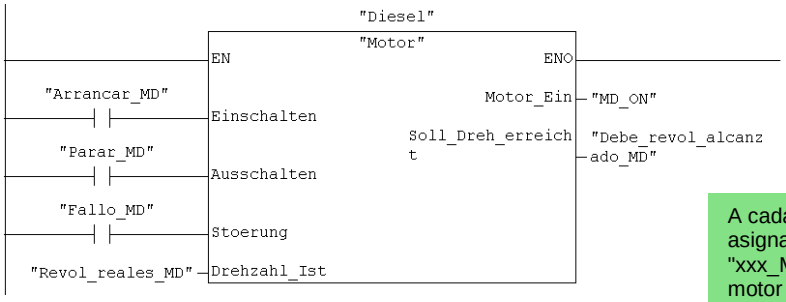
Haga clic en los signos de interrogación y direccione, con ayuda de la lista desplegable, los demás parámetros del bloque de función con los nombres simbólicos que les correspondan.



En el FB "Motor" se muestran las variables de entrada y salida específicas del motor (declaración "in" y "out").

A cada variable se le asignará una señal "xxx_MG" para el motor de gasolina.

Utilizando las direcciones de la lista desplegable programe en otro segmento la llamada del bloque de datos "Diesel" (DB2) desde el bloque de función "Motor" (FB1).



A cada variable se le asignará una señal "xxx_MD" para el motor diesel.

Guarde el programa y cierre el bloque.

Si crea partes del programa con OBs, FBs y DBs, deberá programar la llamada de un bloque subordinado (p.ej., al FB1) en el bloque de orden superior (p.ej., en el OB1). El procedimiento a seguir es siempre el mismo.

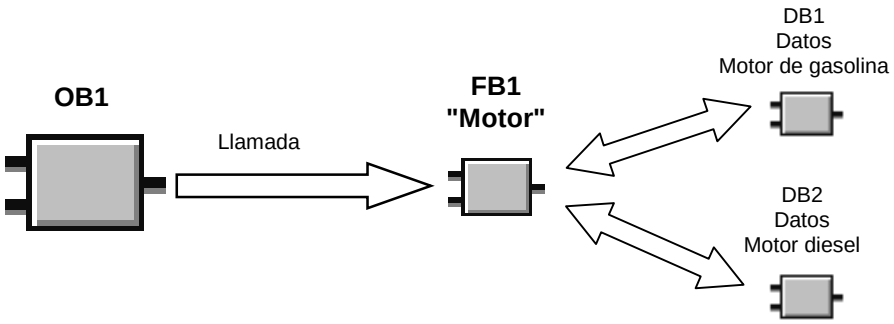
En la tabla de símbolos también puede asignar nombres simbólicos a los bloques (p.ej., el nombre "Motor" al FB1 y el nombre "Gasolina" al DB1).

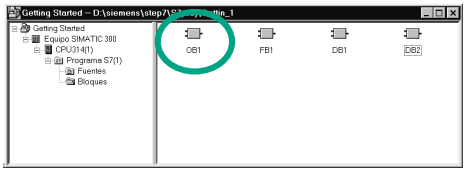
Los bloques programados se pueden archivar o imprimir en todo momento. Estas funciones las encontrará en el Administrador SIMATIC bajo los comandos de menú **Archivo > Archivar** e **Archivo > Imprimir**, respectivamente.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo acceder a las ayudas de referencia", "Descripción del lenguaje KOP" y "Operaciones de control del programa".

5.7 Programar la llamada del bloque en AWL

La programación realizada en un FB no sirve de nada si este FB no es llamado por el OB1. Cada vez que se llama al FB se utiliza un bloque de datos distinto, controlándose así ambos motores.





El proyecto "Getting Started" está abierto en el Administrador SIMATIC.

Navegue hasta la carpeta **Bloques** y abra el OB1.

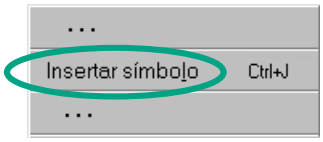
Inserte el segmento 4 en la ventana KOP/AWL/FUP.

```
CALL  "Motor" , "Gasolina"
 Einschalten        :=
 Ausschalten        :=
 Stoerung           :=
 Drehzahl_Ist       :=
 Motor_Ein          :=
 Soll_Dreh_erreicht:=
```

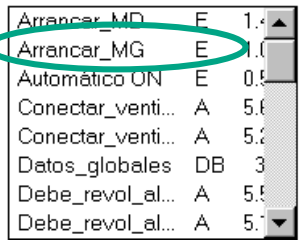
En el área de instrucciones escriba **CALL "Motor", "Gasolina"** y, a continuación, pulse la tecla **Entrar**.

Así se mostrarán todos los parámetros del bloque de función "Gasolina".

Posicione el cursor detrás del signo '=' de 'Arrancar' y pulse la tecla derecha del ratón.



Abra el menú contextual que aparece al pulsar la tecla derecha del ratón y elija **Insertar símbolo**. Entonces se abrirá una lista desplegable. La primera vez tardará un poco más en abrirse.



Haga clic en el nombre **Arrancar_MG**. Así aparecerá escrito entre comillas en el cuadro de entrada.

```
CALL  "Motor" , "Gasolina"
 Einschalten       :="Arrancar_MG"
 Ausschalten       :="Parar_MG"
 Stoerung          :="Fallo_MG"
 Drehzahl_Ist      :="Revol_reales_MG"
 Motor_Ein         :="MG_ON"
 Soll_Dreh_erreicht:="Debe_revol_alcanzado_MG"
```

Con ayuda de la lista desplegable asigne las direcciones necesarias a las variables del bloque de función.

A cada variable se le asignará una señal "xxx_MG" para el motor de gasolina.

```
CALL  "Motor" , "Diesel"
 Einschalten       :="Arrancar_MD"
 Ausschalten       :="Parar_MD"
 Stoerung          :="Parar_MD"
 Drehzahl_Ist      :="Revol_reales_MD"
 Motor_Ein         :="MD_ON"
 Soll_Dreh_erreicht:="Debe_revol_alcanzado_MD"
```

En otro segmento programe la llamada del bloque de función "Motor" (FB1) al bloque de datos "Diesel" (DB2). Proceda como hemos hecho anteriormente.

Guarde el programa y cierre el bloque.

Si crea partes del programa con OBs, FBs y DBs, deberá programar la llamada a un bloque subordinado (p.ej., al FB1) en el bloque de orden superior (p.ej., en el OB1). El procedimiento a seguir es siempre el mismo.

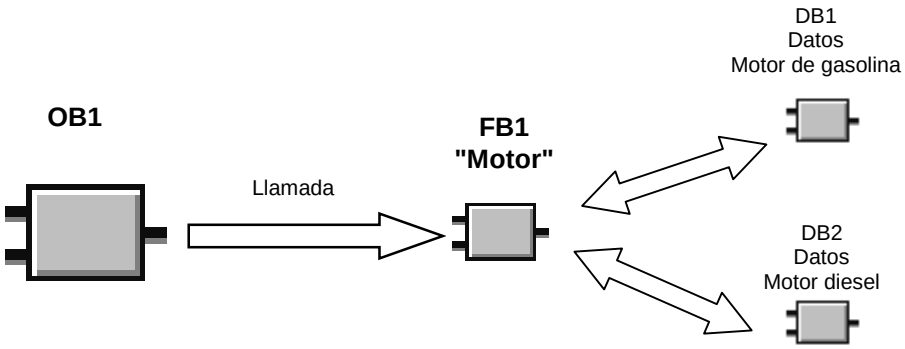
En la tabla de símbolos también puede asignar nombres simbólicos a los bloques (p.ej., el nombre "Motor" al FB1 y el nombre "Gasolina" al DB1).

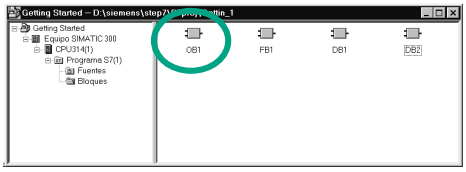
Los bloques programados se pueden archivar o imprimir en todo momento. Estas funciones las encontrará en el Administrador SIMATIC bajo los comandos de menú **Archivo > Archivar** e **Archivo > Imprimir**, respectivamente.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo acceder a las ayudas de referencia", "Descripción del lenguaje AWL" y "Operaciones de conrol del programa".

5.8 Programar la llamada del bloque en FUP

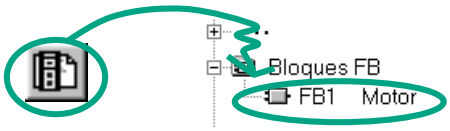
La programación realizada en un FB no sirve de nada si este FB no es llamado por el OB1. Cada vez que se llama al FB se utiliza un bloque de datos distinto, controlándose así ambos motores.



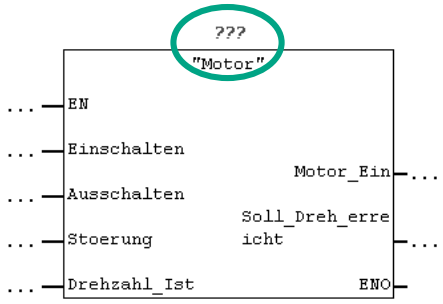


El proyecto "Getting Started" está abierto en el Administrador SIMATIC.

Navegue hasta la carpeta **Bloques** y abra el **OB1**.

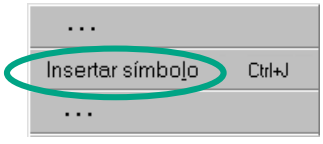


Inserte el segmento 4 en la ventana KOP/AWL/FUP del programa. A continuación navegue por el catálogo de elementos del programa hasta el **FB1** e insértelo.

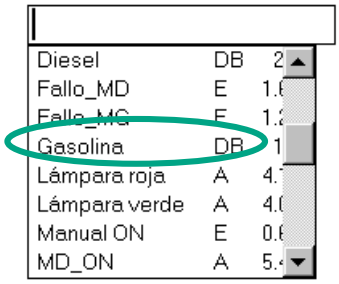


Se mostrarán todas las variables de entrada y salida del motor.

Haga clic en **???** encima de "Motor" y seguidamente, con la tecla derecha del ratón, en el cuadro de entrada.

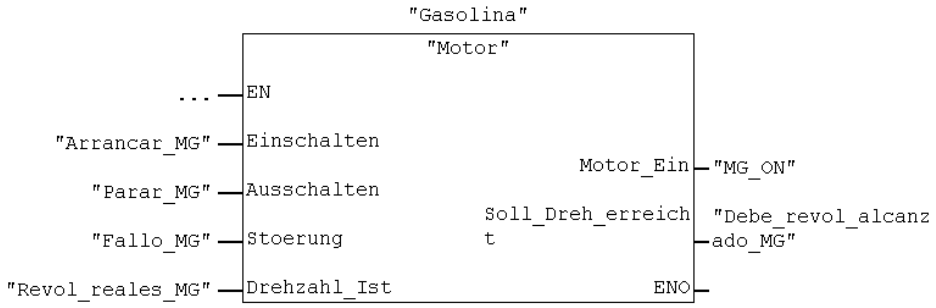


En el menú contextual que aparece al pulsar la tecla derecha del ratón, elija **Insertar símbolo**. Entonces se abrirá una lista desplegable. La primera vez tardará un poco más en abrirse.



Haga clic en el bloque de datos **Gasolina**. Así aparecerá escrito entre comillas en el cuadro de entrada.

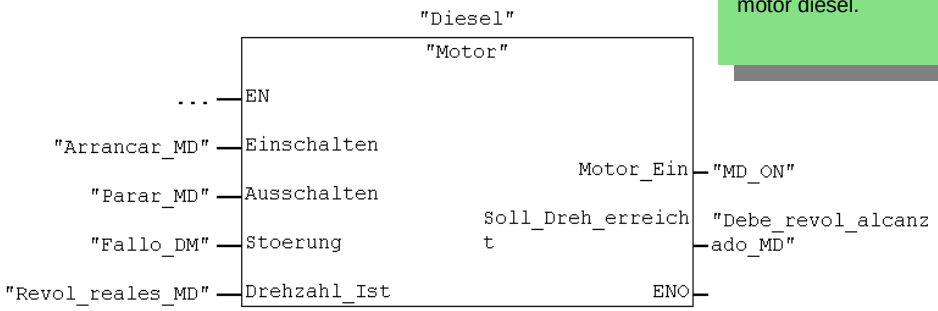
Con ayuda de la lista desplegable direccione los demás parámetros del bloque de función.



A cada variable se le asignará una señal "xxx_MG" para el motor de gasolina.

En otro segmento, programe la llamada del bloque de datos "Diesel" (DB2) desde el bloque de función "Motor" (FB1) utilizando las direcciones de la lista desplegable.

A cada variable se le asignará una señal "xxx_MD" para el motor diesel.



Guarde el programa y cierre el bloque.

Si crea partes del programa con OBs, FBs y DBs, deberá programar la llamada a un bloque subordinado (p.ej., al FB1) en el bloque de orden superior (p.ej., en el OB1). El procedimiento a seguir es siempre el mismo.

En la tabla de símbolos también puede asignar nombres simbólicos a los bloques (p.ej., el nombre "Motor" al FB1 y el nombre "Gasolina" al DB1).

Los bloques programados se pueden archivar o imprimir en todo momento. Estas funciones las encontrará en el Administrador SIMATIC bajo los comandos de menú **Archivo > Archivar** e **Archivo > Imprimir**, respectivamente.

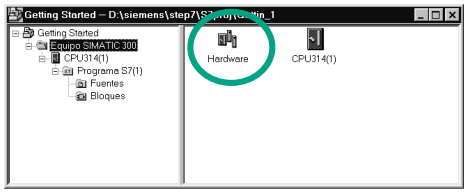
Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo acceder a las ayudas de referencia", "Descripción del lenguaje FUP" y "Operaciones de control del programa".

6 Configurar los módulos centrales

6.1 Configurar el hardware

Para configurar el hardware se tiene que haber creado un proyecto con un equipo SIMATIC. La estructura del proyecto que hemos creado con el "Asistente de STEP 7" en el apartado 2.1 dispone de todos los requisitos necesarios.

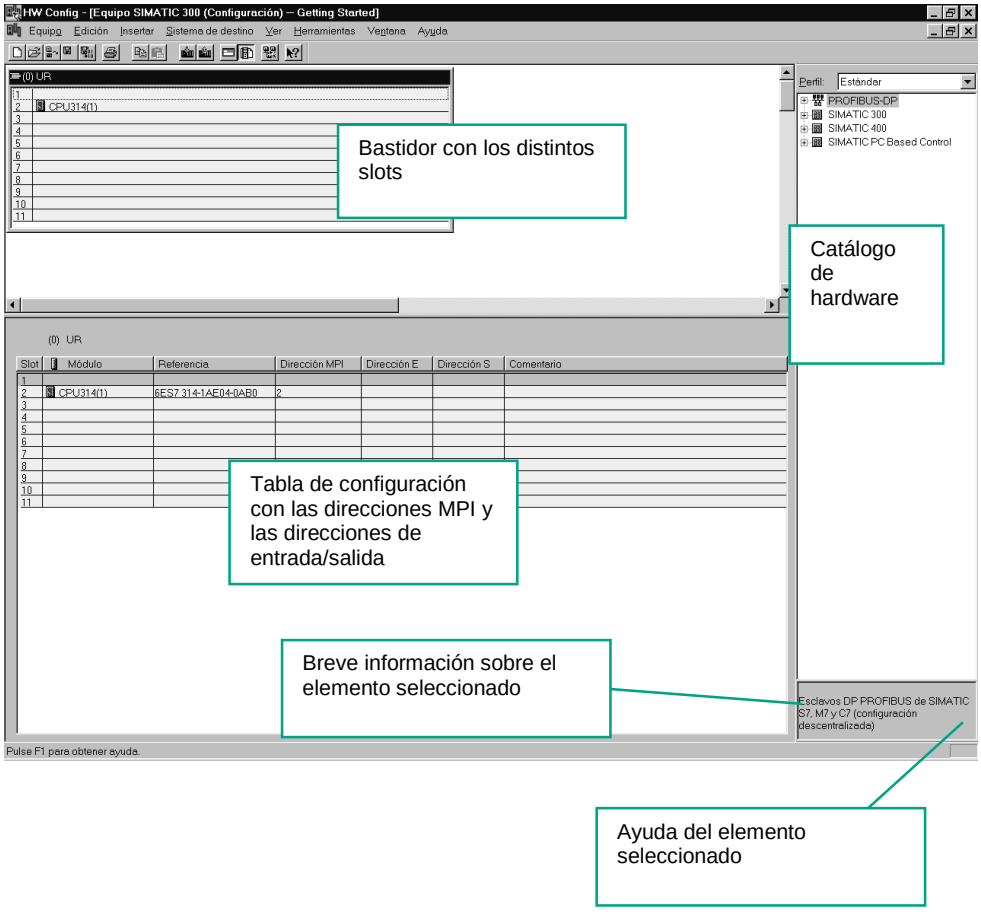
El hardware se configura con STEP 7. Los datos de configuración se transferirán al sistema de automatización posteriormente, al "cargar" (v. apt. 7).

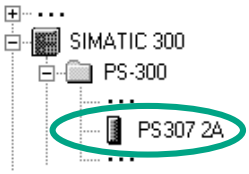


El proyecto "Getting Started" tiene que estar abierto en el Administrador SIMATIC.

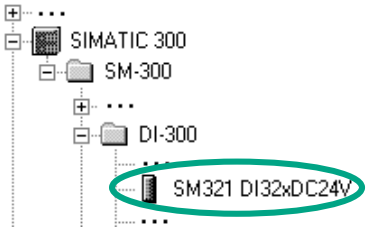
Abra la carpeta **Equipo SIMATIC 300** y haga doble clic en el icono **Hardware**.

Entonces se abrirá la ventana "HW Config" y se visualizará la CPU seleccionada al crear el proyecto. En "Getting Started", se trata de la CPU314.

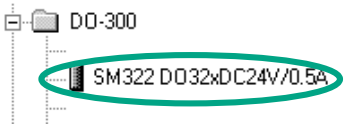

HW Config - [Equipo SIMATIC 300 (Configuración) -- Getting Started]
Equipo Edición Insertar Sistema de destino Ver Herramientas Ventana Ayuda
(0) UR
CPU314(1)
Perfil: Estándar
PROFIBUS-DP
SIMATIC 300
SIMATIC 400
SIMATIC PC Based Control
Slot Módulo Referencia Dirección MPI Dirección E Dirección S Comentario
6ES7 314-1AE04-0AB0
Esclavos DP PROFIBUS de SIMATIC S7, M7 y C7 (configuración descentralizada)
Pulse F1 para obtener ayuda.
Bastidor con los distintos slots
Catálogo de hardware
Tabla de configuración con las direcciones MPI y las direcciones de entrada/salida
Breve información sobre el elemento seleccionado
Ayuda del elemento seleccionado



Lo primero que se necesita es una fuente de alimentación. Navegue por el catálogo hasta la **PS307 2A** e insértela en el slot 1 mediante "Drag and Drop" (arrastrar y soltar).



Navegue por los módulos de entrada (DI, Digital Input) hasta el **SM321 DI32xDC24V** e insértelo en el slot 4. El slot 3 queda vacío.

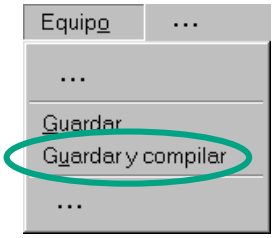


Del mismo modo inserte el módulo de salidas **SM322 DO32xDC24V/0.5A** en el slot 5.

Si desea modificar los parámetros (p.ej., la dirección) de un módulo dentro de un proyecto, abra el módulo en cuestión con un doble clic. Pero primero debe saber los efectos que pueden tener estos cambios en el autómata.

Nuestro proyecto "Getting Started" no requiere modificaciones.

Slot	Módulo	Referencia	Dirección MPI	Dirección E	Dirección S	Comentario
1	PS307 2A	6ES7 307-1BA00-0AA0				
2	CPU314(1)	6ES7 314-1AE04-0AB0	2			
3						
4	DI32xDC24V	6ES7 321-1BL00-0AA0		0...3		
5	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0			4...7	
6						
7						
8						
9						
10						
11						



Con **Guardar y compilar** se preparan los datos para transferirlos a la CPU.

Al salir de "HW Config" se visualiza el icono 'Datos del sistema' en la carpeta 'Bloques'.

Además, con la función **Equipo > Comprobar coherencia** podrá comprobar si la configuración contiene errores. En caso afirmativo, STEP 7 le propondrá algunas soluciones posibles.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Configurar el hardware" y "Configurar módulos centrales".

7 Cargar y probar el programa

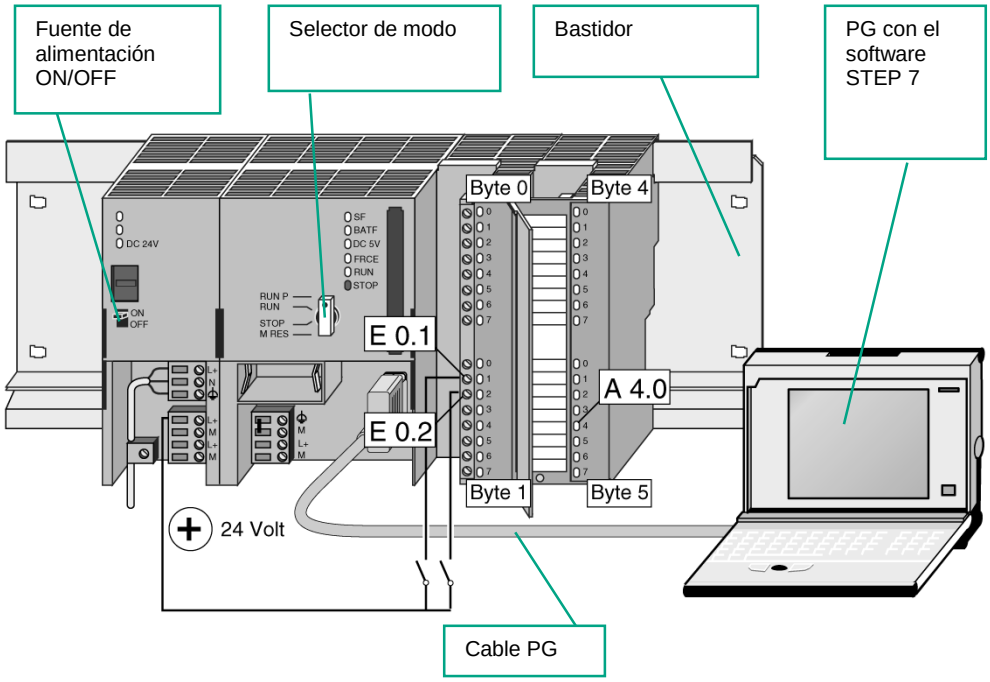
7.1 Establecer una conexión online

Con ayuda del proyecto "zEs01_05_STEP 7_KOP_1-9" suministrado o bien con nuestro "Getting Started" y un test sencillo vamos a cargar el programa en el sistema de automatización (PLC) para probarlo seguidamente.

Requisitos:

- haber configurado el hardware para el proyecto "Getting Started" (v. cap. 6)
- configurar el hardware siguiendo las instrucciones del manual

Ejemplo de conexión en serie (función U (Y)):
La salida A 4.0 se enciende sólo (en el módulo de salidas digitales se enciende el diodo A 4.0) cuando se oprime el pulsador E 0.1 **y** el pulsador E 0.2 con ayuda de la CPU y los cables necesarios.



Montar el hardware

Para montar un módulo en el perfil soporte:

- enchufe el módulo en el conector del bus
- cuélguelo moviéndolo un poco hacia abajo
- atorníllelo
- monte los demás módulos
- inserte la llave en la CPU después de montar todos los módulos

El test también se puede llevar a cabo con un hardware distinto del que acabamos de mostrar. Lo único que deberá tener en cuenta son las direcciones de las entradas y salidas.

STEP 7 ofrece varias posibilidades para realizar el test; p.ej.: observando el programa (status) o con la tabla de variables.

Para más información acerca del montaje de los módulos centrales, consulte los manuales "S7-300 – Configuración, instalación y datos de las CPU" o“S7-400, M7-400 – Configuración e instalación"

7.2 Cargar el programa en el sistema de destino

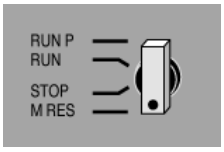
Para poder cargar el programa se tiene que haber establecido la conexión online.

Aplicar tensión

Conecte la red poniendo el interruptor ON/OFF a 'ON'. Se encenderá el diodo "DC 5V" de la CPU.

Gire el selector de modo hasta que llegue a la posición 'STOP' (si no está ya en 'STOP'). El LED "STOP" rojo se encenderá.

Borrar totalmente la CPU y ponerla a RUN



Gire el selector de modo hasta que llegue a la posición **MRES** y manténgalo en esta posición unos 3 s como mínimo hasta que el LED "STOP" rojo parpadee lentamente.

Al realizar un borrado total se borran todos los datos de la CPU. Después, la CPU se volverá a encontrar en su estado inicial.

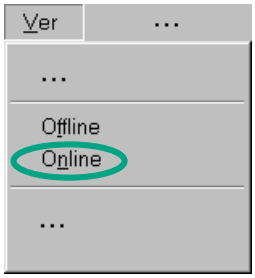
Suelte el selector y, antes de que transcurran 3 s, vuélvalo a poner en la posición **MRES**. Si el LED "STOP" parpadea rápidamente, significa que ha finalizado el borrado total de la CPU.

Si el LED "STOP" no parpadea rápidamente, repita el proceso.

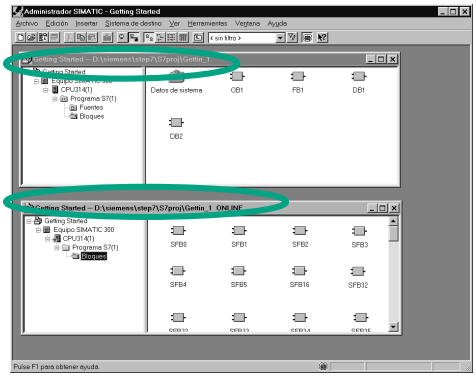
Cargar el programa en la CPU

Para cargar el programa, el selector debe encontrarse en la posición "STOP".

Arranque el Administrador SIMATIC y abra el proyecto "Getting Started" mediante el cuadro de diálogo "Abrir" (si es que aún no está abierto).



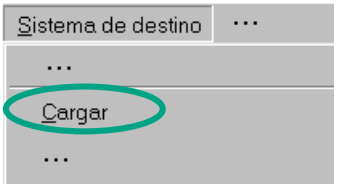
Además de la ventana "Getting Started Offline", abra la ventana "Getting Started Online". El estado offline/online se distingue por el color de la barra de título.



Navegue en ambas ventanas hasta la carpeta **Bloques**.

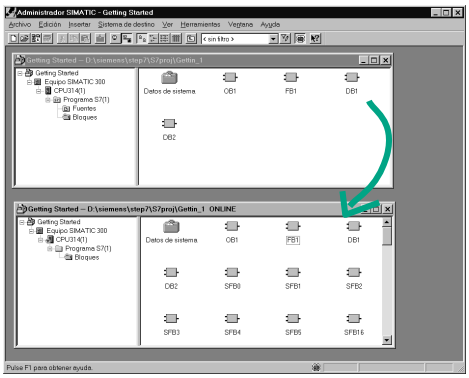
En la ventana "Offline" se muestra el bloque en la PG, mientras que en la ventana "Online" se muestra el bloque en la CPU.

A pesar de haber realizado un borrado total, en la CPU se encuentran las funciones del sistema (SFCs). Estas funciones dependen de la CPU. No se tienen que cargar, pero tampoco se pueden borrar.



Seleccione la carpeta **Bloques** en la ventana "Offline" y, a continuación, cargue el programa en la CPU eligiendo el comando **Sistema de destino > Cargar**.

Confirme con **Aceptar**.



En la ventana "Online" se visualizarán los bloques del programa después de la operación de carga.

El comando de menú **Sistema de destino > Cargar** también se puede activar con el botón correspondiente de la barra de herramientas o el menú contextual que aparece al pulsar la tecla derecha del ratón.

Conectar la CPU y comprobar el estado operativo

Gire el selector de modo hasta la posición **RUN-P**. Entonces se encenderá el LED "RUN" verde y se apagará el LED "STOP" rojo. La CPU está lista para el servicio.

Cuando el LED verde está encendido se puede realizar el test del programa.

Si el LED rojo no cambia a verde, significa que se ha producido un error. Para diagnosticar este error se deberá evaluar el búfer de diagnóstico.

Cargar bloques de uno en uno

Para poder reaccionar rápidamente a los errores que se presenten, se han configurado los bloques de tal modo que se puedan transferir uno a uno a la CPU mediante 'Drag and Drop'.

Al cargar bloques, el selector de la CPU debe estar en "RUN-P" o "STOP". Los bloques cargados en el estado operativo "RUN" se activan inmediatamente. No olvide:

- Si se sobreescriben bloques correctos con bloques que contienen errores, puede producirse una avería en la instalación. Para evitarlo, pruebe los bloques antes de cargarlos.
- Si no se ha seguido el orden correcto al cargar los bloques – primero los niveles inferiores, y después los superiores –, la CPU pasará al estado operativo "STOP". Para evitarlo, cargue todo el programa en la CPU.

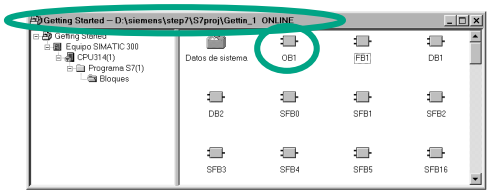
Programar en modo online

En algunos casos puede ser interesante para el test modificar los bloques que ya se han cargado en la CPU. Para ello haga doble clic en el bloque deseado en la ventana "Online", así se abrirá la ventana KOP/AWL/FUP. A continuación programe el bloque del modo habitual. No olvide que, una vez programado, se activará inmediatamente en la CPU.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cargar" y "Cargar en el sistema de destino desde la PG".

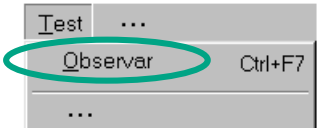
7.3 Test visualizando el estado del programa

Con la función 'Status' (observar el programa) se puede probar el programa de un bloque. Para ello tiene que haberse establecido una conexión online con la CPU, la CPU tiene que estar en RUN o RUN-P y el programa tiene que haberse cargado en la CPU.



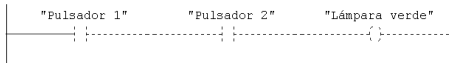
Abra el **OB1** en la ventana "Getting Started Online" del programa.

Se abrirá la ventana KOP/AWL/FUP.



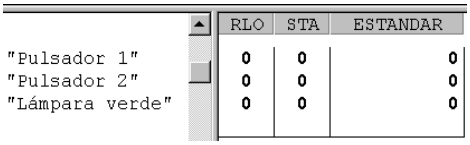
Active la función **Test > Observar**.

Probar el programa con KOP



Se visualiza la conexión en serie del segmento 1 en KOP. Hasta el pulsador 1 (E 0.1), el circuito se representa mediante una línea continua, es decir, se aplica tensión hasta el pulsador 1.

Probar el programa con AWL

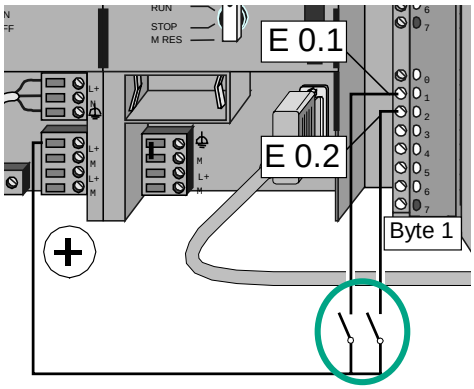


En AWL los
– resultados lógicos (RLO),
– bits de estado (STA) y
– estados estándar (STANDARD)
se visualizan en forma de tabla.

Probar el programa con FUP

El estado de señal se marca con un "0" o un "1". La línea discontinua significa que no hay resultado lógico.

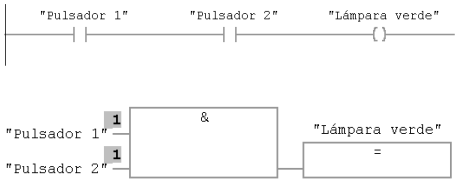
Con el comando **Herramientas > Preferencias** se puede cambiar el modo de visualización del lenguaje de programación durante el test.



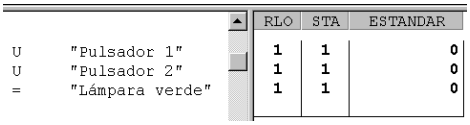
Ahora cierre los dos contactos en el autómata preparado para el test.

En el módulo de entradas se encienden los diodos de las entradas E 0.1 y E 0.2.

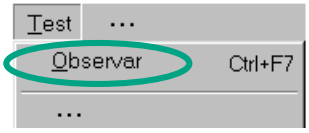
En el módulo de salidas, el diodo de la salida A 4.0.



En los lenguajes de programación KOP / FUP puede seguir el test fijándose en el cambio de color del segmento programado. El color cambia cuando se cumple el resultado lógico.



En el lenguaje de programación AWL, cambia el contenido de las columnas STA y RLO si se cumple el resultado lógico.



Desactive el comando **Test > Observar** y cierre la ventana.

A continuación cierre la ventana "Online" en el Administrador SIMATIC.

Si el tamaño del programa es considerable, recomendamos no cargarlo entero en la CPU y ejecutarlo allí, pues el diagnóstico de errores se complica tanto más cuanto mayor sea el tamaño del programa. Es mucho mejor cargar y probar los bloques de uno en uno.

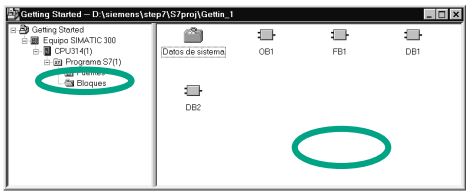
Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Test" y "Test con el estado del programa (status)".

7.4 Probar el programa con la tabla de variables

El test de las variables del programa consiste en observar y forzar dichas variables. Para poder realizar este test tiene que existir una conexión online con la CPU, ésta tiene que estar en RUN-P y el programa se tiene que haber cargado ya.

Del mismo modo que en el test observando el estado del programa, al probar las variables puede observar las entradas y salidas del segmento 1 (conexión en serie o función U (Y)) en la tabla de variables. Además, ajustando un número REAL de revoluciones puede probar el comparador previsto para el número de revoluciones del motor en el FB1.

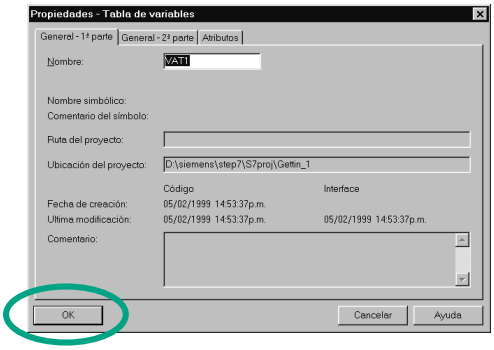
Crear la tabla de variables



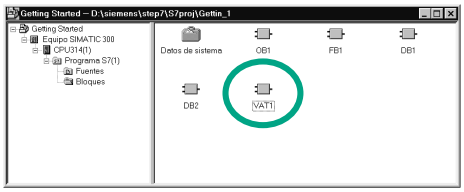
El proyecto "Getting Started Offline" está abierto en el Administrador SIMATIC.

Navegue hasta la carpeta **Bloques** y haga clic con la tecla derecha del ratón en la mitad derecha de la ventana.

Inserte la **tabla de variables** del menú contextual que aparece al pulsar la tecla derecha del ratón.



Acepte los ajustes estándar cerrando el cuadro de diálogo "Propiedades" con **Aceptar**.

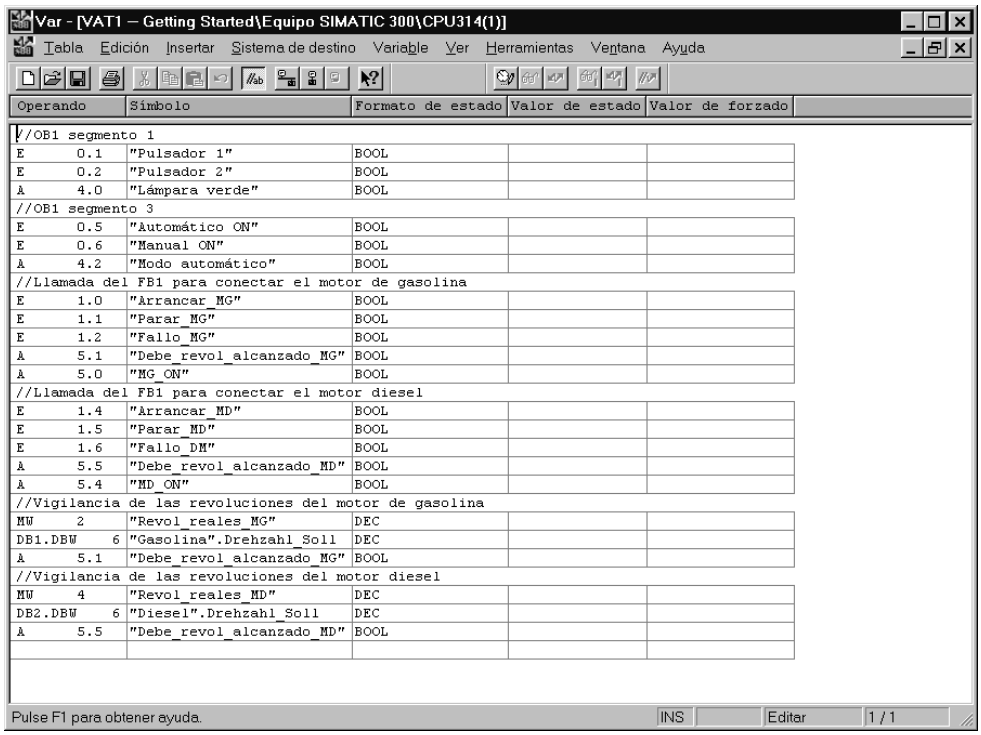


Se creará una VAT1 (tabla de variables) en la carpeta 'Bloques'.

Abra la **VAT1** (con un doble clic) y se abrirá la ventana "Observar y forzar variable".

Primero la tabla de variables está vacía. Para el ejemplo "Getting Started" introduzca los nombres simbólicos o los operandos representados en la figura. Los demás datos se introducirán automáticamente al confirmarlos con 'Entrar'.

Cambie el formato de estado de todos los valores de revoluciones poniéndolo a DEC. Para ello haga clic en la celda correspondiente (la flecha del ratón cambia de forma cuando se encuentra en la columna 'Formato de estado') y seleccione el formato DEC.


Var - [VAT1 – Getting Started\Equipo SIMATIC 300\CPU314(1)]
Tabla Edición Insertar Sistema de destino Variable Ver Herramientas Ventana Ayuda
Operando | Símbolo | Formato de estado | Valor de estado | Valor de forzado
//OB1 segmento 1
E 0.1 "Pulsador 1" BOOL
E 0.2 "Pulsador 2" BOOL
A 4.0 "Lámpara verde" BOOL
//OB1 segmento 3
E 0.5 "Automático ON" BOOL
E 0.6 "Manual ON" BOOL
A 4.2 "Modo automático" BOOL
//Llamada del FB1 para conectar el motor de gasolina
E 1.0 "Arrancar_MG" BOOL
E 1.1 "Parar_MG" BOOL
E 1.2 "Fallo_MG" BOOL
A 5.1 "Debe_revol_alcanzado_MG" BOOL
A 5.0 "MG_ON" BOOL
//Llamada del FB1 para conectar el motor diesel
E 1.4 "Arrancar_MD" BOOL
E 1.5 "Parar_MD" BOOL
E 1.6 "Fallo_DM" BOOL
A 5.5 "Debe_revol_alcanzado_MD" BOOL
A 5.4 "MD_ON" BOOL
//Vigilancia de las revoluciones del motor de gasolina
MW 2 "Revol_reales_MG" DEC
DB1.DBW 6 "Gasolina".Drehzahl_Soll DEC
A 5.1 "Debe_revol_alcanzado_MG" BOOL
//Vigilancia de las revoluciones del motor diesel
MW 4 "Revol_reales_MD" DEC
DB2.DBW 6 "Diesel".Drehzahl_Soll DEC
A 5.5 "Debe_revol_alcanzado_MD" BOOL
Pulse F1 para obtener ayuda.
INS
Editar
1 / 1

Guarde la tabla de variables.

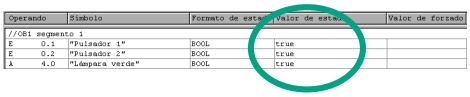
Conectar la tabla de variables online

Haga clic en la ventana "Observar y forzar variable" en **ON** para establecer la conexión con la CPU configurada. En la barra de estado aparecerá "Online".

Posicione el selector de modo de la CPU en **RUN-P** (si aún no lo está).

Observar variables

Haga clic en **Observar variable**. En la barra de estado se visualizará el funcionamiento de la CPU.

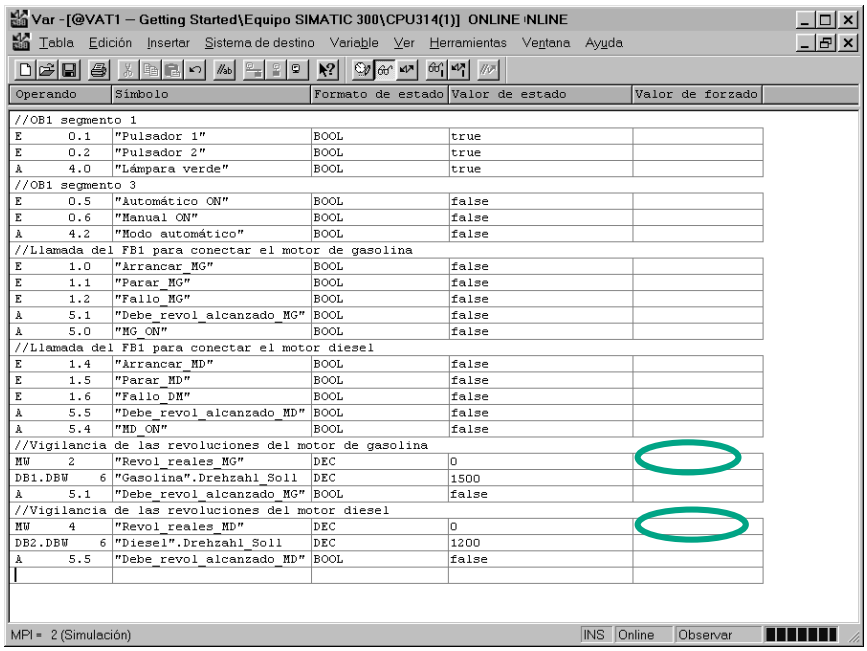


Oprima los pulsadores 1 y 2 y observe el resultado en la tabla de variables.

Verá que los valores de estado de la tabla cambian de 'false' a 'true'.

Forzar variables

En la columna 'Valor de forzado' introduzca el valor "1500" para el operando MW2 y "1300" para el operando MW4.

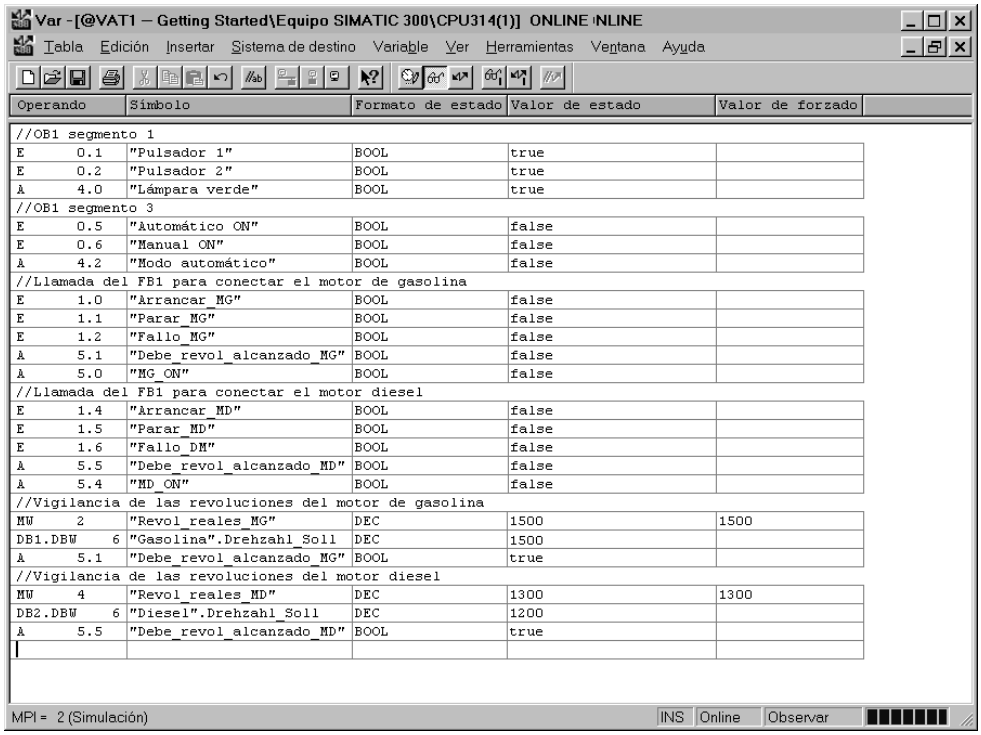


Operando	Símbolo	Formato de estado	Valor de estado	Valor de forzado
//OB1 segmento 1				
E 0.1	"Pulsador 1"	BOOL	true	
E 0.2	"Pulsador 2"	BOOL	true	
A 4.0	"Lámpara verde"	BOOL	true	
//OB1 segmento 3				
E 0.5	"Automático ON"	BOOL	false	
E 0.6	"Manual ON"	BOOL	false	
A 4.2	"Modo automático"	BOOL	false	
//Llamada del FB1 para conectar el motor de gasolina				
E 1.0	"Arrancar_MG"	BOOL	false	
E 1.1	"Parar_MG"	BOOL	false	
E 1.2	"Fallo_MG"	BOOL	false	
A 5.1	"Debe_revol_alcanzado_MG"	BOOL	false	
A 5.0	"MG_ON"	BOOL	false	
//Llamada del FB1 para conectar el motor diesel				
E 1.4	"Arrancar_MD"	BOOL	false	
E 1.5	"Parar_MD"	BOOL	false	
E 1.6	"Fallo_DM"	BOOL	false	
A 5.5	"Debe_revol_alcanzado_MD"	BOOL	false	
A 5.4	"MD_ON"	BOOL	false	
//Vigilancia de las revoluciones del motor de gasolina				
MW 2	"Revol_reales_MG"	DEC	0	
DB1.DBW 6	"Gasolina".Drehzahl_Soll	DEC	1500	
A 5.1	"Debe_revol_alcanzado_MG"	BOOL	false	
//Vigilancia de las revoluciones del motor diesel				
MW 4	"Revol_reales_MD"	DEC	0	
DB2.DBW 6	"Diesel".Drehzahl_Soll	DEC	1200	
A 5.5	"Debe_revol_alcanzado_MD"	BOOL	false	

Transfiera los valores de estado a la CPU.

Una vez transferidos, se procesarán estos valores en la CPU y se visualizará el resultado de la comparación.

Finalice la observación de las variables y cierre la ventana. Si aparece un mensaje en la pantalla, confirme con **Sí** o con **Aceptar**.


Var - [@VAT1 -- Getting Started\Equipo SIMATIC 300\CPU314(1)] ONLINE INLINE
Tabla Edición Insertar Sistema de destino Variable Ver Herramientas Ventana Ayuda
MPI = 2 (Simulación)
INS Online Observar

Operando	Símbolo	Formato de estado	Valor de estado	Valor de forzado
//OB1 segmento 1				
E 0.1	"Pulsador 1"	BOOL	true	
E 0.2	"Pulsador 2"	BOOL	true	
A 4.0	"Lámpara verde"	BOOL	true	
//OB1 segmento 3				
E 0.5	"Automático ON"	BOOL	false	
E 0.6	"Manual ON"	BOOL	false	
A 4.2	"Modo automático"	BOOL	false	
//Llamada del FB1 para conectar el motor de gasolina				
E 1.0	"Arrancar_MG"	BOOL	false	
E 1.1	"Parar_MG"	BOOL	false	
E 1.2	"Fallo_MG"	BOOL	false	
A 5.1	"Debe_revol_alcanzado_MG"	BOOL	false	
A 5.0	"MG_ON"	BOOL	false	
//Llamada del FB1 para conectar el motor diesel				
E 1.4	"Arrancar_MD"	BOOL	false	
E 1.5	"Parar_MD"	BOOL	false	
E 1.6	"Fallo_DM"	BOOL	false	
A 5.5	"Debe_revol_alcanzado_MD"	BOOL	false	
A 5.4	"MD_ON"	BOOL	false	
//Vigilancia de las revoluciones del motor de gasolina				
MW 2	"Revol_reales_MG"	DEC	1500	1500
DB1.DBW 6	"Gasolina".Drehzahl_Soll	DEC	1500	
A 5.1	"Debe_revol_alcanzado_MG"	BOOL	true	
//Vigilancia de las revoluciones del motor diesel				
MW 4	"Revol_reales_MD"	DEC	1300	1300
DB2.DBW 6	"Diesel".Drehzahl_Soll	DEC	1200	
A 5.5	"Debe_revol_alcanzado_MD"	BOOL	true	

Generalmente, las tablas de variables de gran tamaño no se ven completas en la pantalla.

En este caso recomendamos crear -con ayuda de STEP 7- varias tablas para un mismo programa S7, pudiéndolas adaptar al test que desee realizar.

Del mismo modo que hacemos con los bloques, también se pueden asignar nombres a las tablas de variables (p.ej., el nombre 'OB1_Segmento1' a la tabla VAT1). Estas asignaciones se llevan a cabo con la tabla de símbolos.

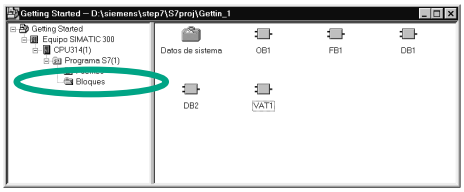
Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Test" y "Test con la tabla de variables".

7.5 Evaluar el búfer de diagnóstico

En el caso excepcional que la CPU pase a 'STOP' al ejecutar un programa S7 o que no se pueda conmutar a 'RUN' después de cargar el programa, puede averiguar la causa del error consultando los eventos listados en el búfer de diagnóstico.

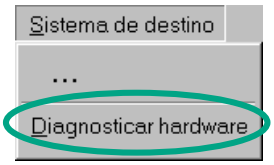
Para que ello sea posible tiene que existir una conexión online con la CPU y ésta tiene que estar en el estado operativo "STOP".

Primero gire el selector de modo de la CPU a "STOP".

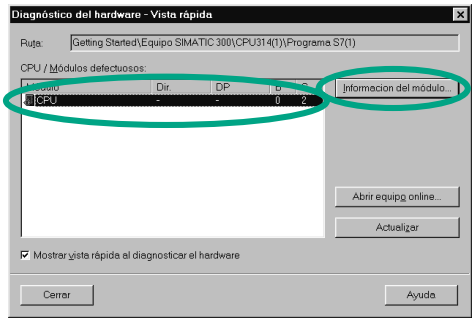


El proyecto "Getting Started Offline" tiene que estar abierto en el Administrador SIMATIC.

Seleccione la carpeta **Bloques**.



Si su proyecto dispone de varias CPUs, averigüe cuál de ellas ha pasado a 'STOP'.



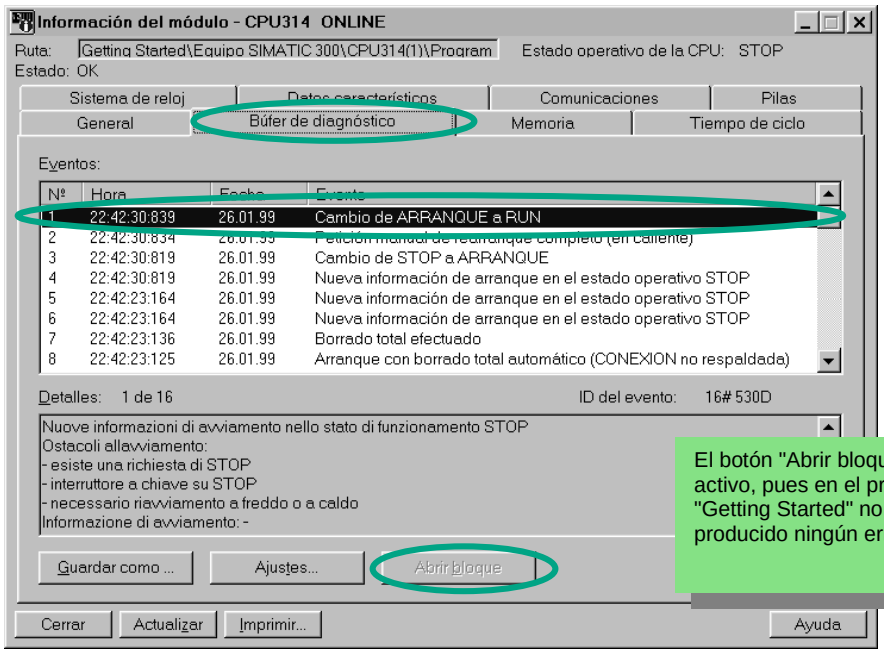
En el cuadro de diálogo "Diagnóstico del hardware" se visualizarán las CPUs accesibles, seleccionándose la que haya pasado a 'STOP'.

El proyecto "Getting Started" sólo dispone de una CPU, por lo que ésta será la única que se visualizará.

Para evaluar el búfer de diagnóstico de esta CPU haga clic en **Información del módulo**.

Si sólo se ha conectado una CPU, también se puede consultar directamente su estado eligiendo el comando de menú **Sistema de destino > Información del módulo**.

En la ventana "Información del módulo" se obtienen informaciones acerca de las propiedades y los parámetros de la CPU. Para averiguar la causa del 'STOP' de la CPU seleccione la ficha **Búfer de diagnóstico**.


Información del módulo - CPU314 ONLINE
Ruta: Getting Started\Equipo SIMATIC 300\CPU314(1)\Program
Estado operativo de la CPU: STOP
Estado: OK
Sistema de reloj
Datos característicos
Comunicaciones
Pilas
General
Búfer de diagnóstico
Memoria
Tiempo de ciclo
Eventos:
Nº Hora Fecha Evento
1 22:42:30:839 26.01.99 Cambio de ARRANQUE a RUN
2 22:42:30:834 26.01.99 Petición manual de rearranque completo (en caliente)
3 22:42:30:819 26.01.99 Cambio de STOP a ARRANQUE
4 22:42:30:819 26.01.99 Nueva información de arranque en el estado operativo STOP
5 22:42:23:164 26.01.99 Nueva información de arranque en el estado operativo STOP
6 22:42:23:164 26.01.99 Nueva información de arranque en el estado operativo STOP
7 22:42:23:136 26.01.99 Borrado total efectuado
8 22:42:23:125 26.01.99 Arranque con borrado total automático (CONEXION no respaldada)
Detalles: 1 de 16
ID del evento: 16# 530D
Nuove informazioni di avviamento nello stato di funzionamento STOP
Ostacoli allavviamento:
- esiste una richiesta di STOP
- interruttore a chiave su STOP
- necessario riavviamento a freddo o a caldo
Informazione di avviamento: -
Guardar como ...
Ajustes...
Abrir bloque
Cerrar
Actualizar
Imprimir...
Ayuda

El botón "Abrir bloque" no está activo, pues en el proyecto "Getting Started" no se ha producido ningún error.

Arriba de todo aparece el evento más reciente (n° 1), y se indica la causa del 'STOP'. Cierre todas las ventanas, a excepción del Administrador SIMATIC.

Si la causa del 'STOP' es un error de programación, seleccione el evento y haga clic en el botón **Abrir bloque**.

Entonces el bloque se abrirá en la ventana KOP/AWL/FUP seleccionándose el segmento en el que ha aparecido el error.

Al finalizar este capítulo, habremos seguido -con el programa de ejemplo "Getting Started"- un proyecto desde su creación hasta el test. En los capítulos siguientes profundizaremos algunos aspectos con ayuda de ejercicios más detallados.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Diagnóstico", "Información del módulo" y "Funciones de la información del módulo".

8 Programar una función (FC)

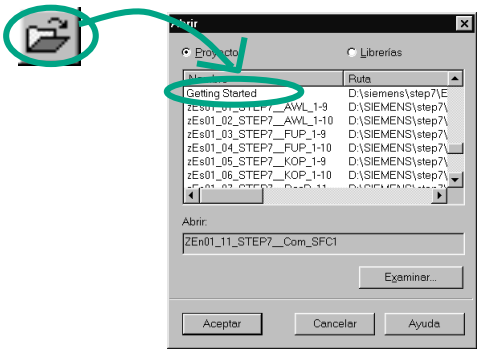
8.1 Crear y abrir la función

La función, como el bloque de función, está subordinada al bloque de organización. Para que pueda ser procesada por la CPU, ésta tiene que ser llamada también desde el bloque de orden superior. Para ello, al contrario que en el caso del bloque de función, no se requiere ningún bloque de datos.

En las funciones, los parámetros también se listan en la tabla de declaración de variables, pero no se admiten datos locales estáticos.

Las funciones, como los bloques de función, se programan en la ventana KOP/FUP/AWL.

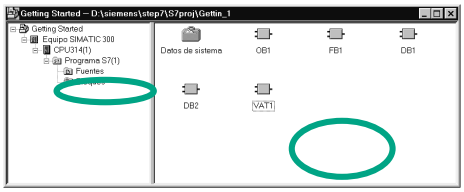
Antes de crear la función tendría que familiarizarse con la programación en KOP, FUP o AWL (v. caps. 4 y 5), así como con la programación simbólica (v. cap. 3).



Si ha creado el proyecto de ejemplo "Getting Started" (caps. 1–7), ábralo.

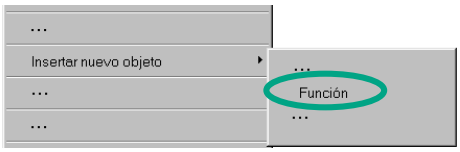
Si no lo ha creado, cree ahora un proyecto en el Administrador SIMATIC con **Archivo > Asistente "Nuevo proyecto"**. Siga las instrucciones del apt. 2.1 y asigne el nombre "Función Getting Started" al nuevo proyecto.

Nosotros nos basaremos en el proyecto "Getting Started". Sin embargo, puede seguir cada uno de los pasos igualmente con otro proyecto.

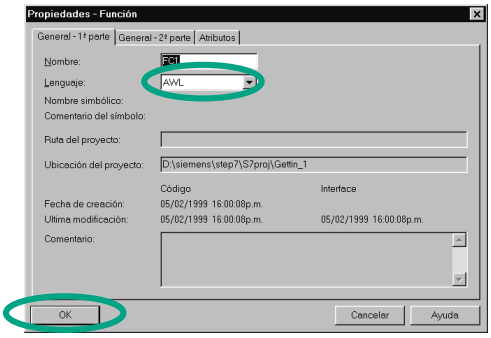


Navegue hasta la carpeta **Bloques** y ábrala.

Haga clic con la tecla derecha del ratón en la mitad derecha de la ventana.

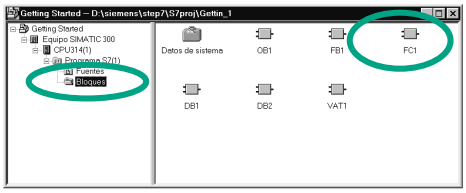


Inserte una **función (FC)** con ayuda del menú contextual.



En el cuadro de diálogo "Propiedades – Función" acepte el nombre 'FC1' y seleccione el lenguaje deseado.

Confirme los demás ajustes con **Aceptar (OK)**.



La función 'FC1' se ha insertado en la carpeta 'Bloques'.

Abra la **FC1** con un doble clic.

Contrariamente a los bloques de función, las funciones no permiten definir datos estáticos en la tabla de declaración de variables.

Los datos estáticos definidos en un bloque de función siguen existiendo después de ejecutar el bloque. Ejemplos de datos estáticos: las marcas utilizadas para los valores límite "Drehzahl" (Revoluciones) (v. cap. 5).

Para programar la función puede utilizar, como es habitual, los nombres simbólicos de la tabla de símbolos.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo diseñar una solución de automatización", "Nociones básicas para diseñar la estructura del programa" y "Bloques en el programa de usuario".

8.2 Programar la función

A continuación programaremos una función de temporización en nuestro ejemplo. Esta función de temporización activará un retardo a la desconexión, p.ej., para que el ventilador que funciona automáticamente al arrancar el motor (v. cap. 5) siga funcionando unos cuatro segundos más tras desconectarse el motor.

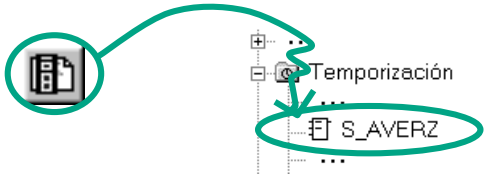
Como hemos mencionado anteriormente, los parámetros de entrada y salida de la función (declaración "in" y "out") deben indicarse en la tabla de declaración de variables.

Se ha abierto la ventana KOP/AWL/FUP. Utilice la tabla de declaración de variables como hemos hecho con la del bloque de función (v. cap. 5).

Haga las declaraciones siguientes.

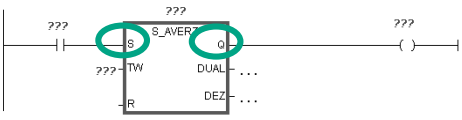
Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
0.0	in	Motor_Ein	BOOL		Señal para la conexión del motor
2.0	in	Zeitfunktion	TIMER		Función de temporización utilizada para el retardo a la desconexión
4.0	out	Luefter_Ein	BOOL		Señal para la conexión del ventilador
	in_out				
	temp				

Programar la función de temporización en KOP



Seleccione el circuito para introducir la instrucción KOP.

Navegue por el catálogo de elementos del programa hasta el elemento **S_AVERZ** (arrancar temporizador de retardo a la desconexión) e insértelo.

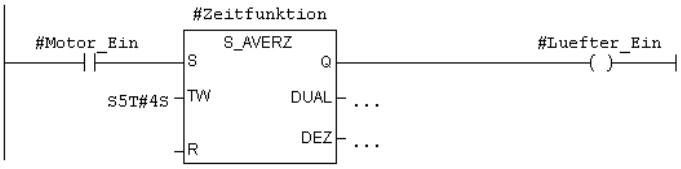


Inserte un contacto normalmente abierto antes de la entrada **S**. Inserte una bobina después de la salida **Q**.

Seleccione los signos de interrogación e introduzca los nombres tal y como aparecen en la tabla de declaración de variables (se marcarán automáticamente con el signo '#').

Ajuste el retardo en la entrada TW del elemento S_AVERZ. Si define 'S5T#4s', significa que una constante del tipo de datos S5Time#(S5T#) tiene una duración de cuatro segundos (4s).

A continuación guarde la función y cierre la ventana.



La "#función de temporización" se activa con el parámetro de entrada "#Motor_ON" (#Motor_Ein). Más tarde, al realizar la llamada del OB1, se le asignarán primero los parámetros del motor de gasolina y después los parámetros del motor diesel (p.ej., T1 para "Retardo_descon_MG"). Posteriormente tendrá que indicar los nombres simbólicos de estos parámetros en la tabla de símbolos.

Programar la función de temporización en AWL

```
U     #Motor_Ein
L     S5T#4S
SA    #Zeitfunktion
U     #Zeitfunktion
=     #Luefter_Ein
```

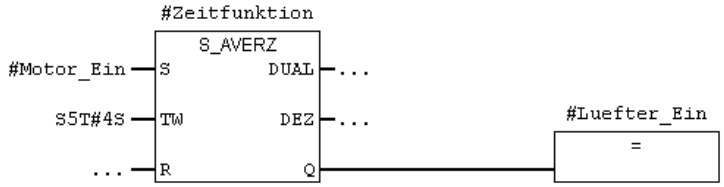
Si programa en AWL, seleccione el campo de entrada que se encuentra debajo del segmento e introduzca la instrucción que le mostramos a la izquierda.

A continuación guarde la función y cierre la ventana.

Programar la función de temporización en FUP

Si programa en FUP, seleccione el campo de entrada que se encuentra debajo del segmento e introduzca el programa FUP de la función de temporización que le mostramos a continuación.

Guarde la función y cierre la ventana.



Para poder procesar la función de temporización, ésta tiene que ser llamada desde el bloque de orden superior (en nuestro ejemplo, por el OB1).

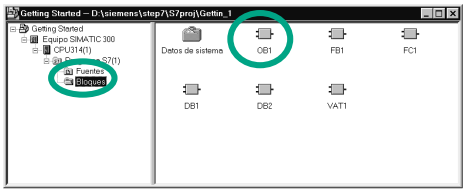
Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo acceder a las ayudas de referencia", "Saltos a descripciones de lenguajes KOP/FUP o AWL" y "Operaciones de temporización".

8.3 Llamada de la función en el OB1

La llamada de la función FC1 en el OB1se realiza del mismo modo que la llamada del bloque de función. Todos los parámetros de la función reciben en el OB1 los valores de los correspondientes operandos del motor de gasolina o del motor diesel.

Estos operandos aún no han sido definidos en la tabla de símbolos, por lo que sus nombres simbólicos deberán indicarse más tarde en la tabla.

Un operando es la parte de una instrucción de STEP 7 que determina con qué debe realizar una acción el procesador. Puede direccionarse de forma absoluta o simbólica.



El proyecto "Getting Started" o el proyecto que acaba de crear está abierto en el Administrador SIMATIC.

Navegue hasta la carpeta **Bloques** y abra el **OB1**.

Se abrirá la ventana KOP/AWL/FUP del programa.

Si en el capítulo 4 ha copiado la tabla de símbolos de un proyecto de ejemplo (zEs01_01_STEP7_AWL_1-9, zEs01_05_STEP7_KOP_1-9, zEs01_04_STEP7_FUP_1-9) en el proyecto "Getting Started", no deberá introducir más símbolos.

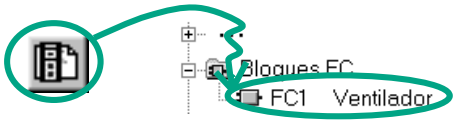
Añadir símbolos a posteriori

Abra la tabla de símbolos eligiendo el comando **Herramientas > Tabla de símbolos**. Desplácese por la barra que se encuentra a la derecha de la ventana hasta llegar a la tabla de símbolos.

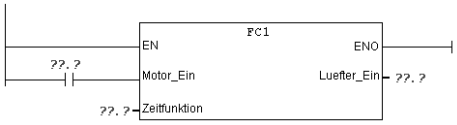
Complete la tabla con los siguientes símbolos.

Símbolo	Dirección	Tipo de datos	Comentario
Retardo_descon_MD	T 2	TIMER	Retardo para el ventilador del motor diesel
Retardo_descon_MG	T 1	TIMER	Retardo para el ventilador del motor de gasolin
Ventilador	FC 1	FC 1	Control del ventilador
Conectar_ventilador_MD	A 5.6	BOOL	Comando para conectar ventilador motor diese
Conectar_ventilador_MG	A 5.2	BOOL	Comando para conectar ventilador motor de ga

Programar la llamada en KOP



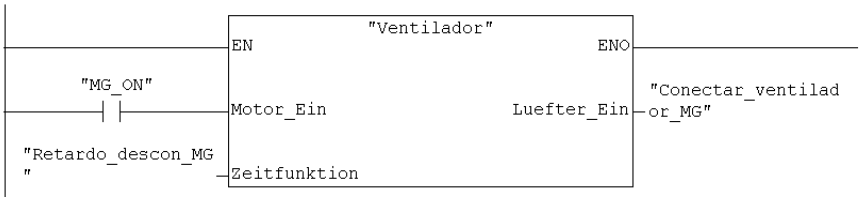
Se encuentra en la vista **KOP**. Inserte otro segmento (n° 6). A continuación navegue por el catálogo de elementos del programa hasta la FC1 e insértela.



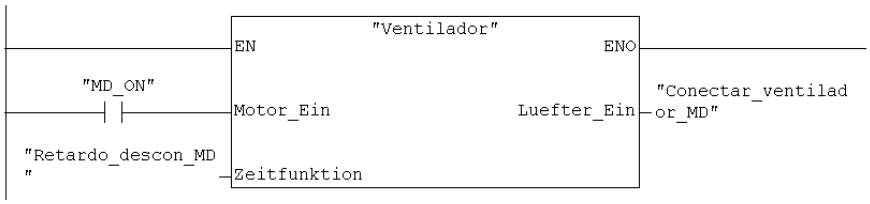
Inserte un contacto normalmente abierto antes de "Motor_ON".

Con el comando de menú **Ver > Mostrar > Representación simbólica** puede cambiar de la representación simbólica a la absoluta, y viceversa.

Haga clic en los signos de interrogación de la llamada de la FC1 e introduzca los nombres simbólicos.



Programe en el segmento 7 la llamada a la función FC1 con el operando del motor diesel. Para ello proceda como hemos hecho con el segmento anterior (los operandos para el motor diesel ya se han registrado en la tabla de símbolos).



Guarde el bloque y cierre la ventana.

Active el comando **Ver > Mostrar > Información del símbolo** para obtener en cada segmento información sobre las distintas direcciones.

Para visualizar varios segmentos en la pantalla, desactive **Ver > Mostrar > Comentario** y, dado el caso, **Ver > Mostrar > Información del símbolo**.

Con **Ver > Factor de zoom** puede modificar el tamaño en el que se representan los segmentos.

Programar la llamada en AWL

```
Segm. 6 : Control del ventilador para motor de gasolina
      CALL  "Ventilador"
       Motor_Ein    :="MG_ON"
       Zeitfunktion:="Retardo_descon_MG"
       Luefter_Ein :="Conectar_ventilador_MG"

Segm. 7 : Control del ventilador para motor diesel
      CALL  "Ventilador"
       Motor_Ein    :="MD_ON"
       Zeitfunktion:="Retardo_descon_MD"
       Luefter_Ein :="Conectar_ventilador_MD"
```

Si programa en AWL, seleccione el campo de entrada que se encuentra debajo de los nuevos segmentos e introduzca las instrucciones AWL que le mostramos a la izquierda.

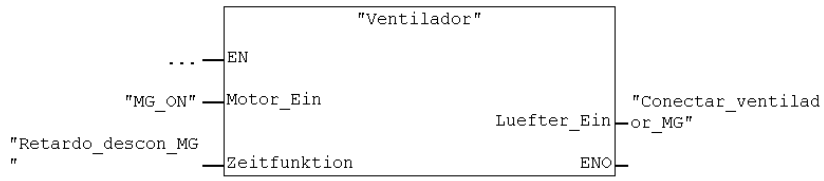
A continuación guarde la llamada y cierre la ventana.

Programar la llamada en FUP

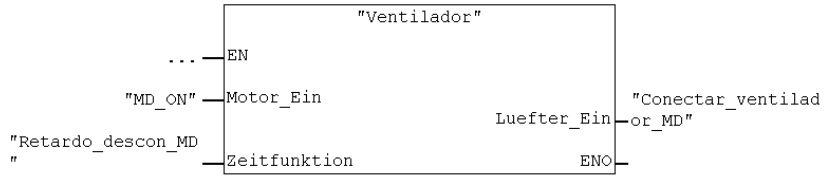
Si programa en FUP, seleccione el campo de entrada que se encuentra debajo de los nuevos segmentos e introduzca las instrucciones que le mostramos a continuación.

Guarde la llamada y cierre la ventana.

Segm. 6 : Control del ventilador para motor de gasolina



Segm. 7 : Control del ventilador para motor diesel



En nuestro ejemplo hemos programado la llamada a las funciones sin condición, es decir, como una llamada que se efectúa siempre.

Dependiendo de la tarea de automatización que esté programando, la llamada de las FCs o de los FBs también se puede combinar con determinadas condiciones, p.ej., en una entrada o en una precombinación. Para programar condiciones disponemos de la entrada EN y de la salida ENO.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo acceder a las ayudas de referencia", "Saltos en descripciones de lenguajes KOP/FUP o AWL" y "Control de programa" o "Operaciones de control del programa".

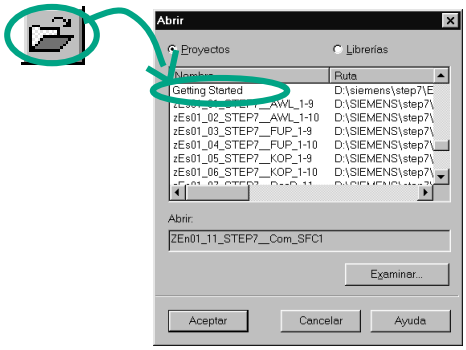
9 Programar un bloque de datos globales

9.1 Crear y abrir el bloque de datos globales

Si el número de marcas internas de una CPU (células de memoria) no es suficiente para todos los datos, se pueden depositar algunos datos en un bloque de datos globales.

Todos los bloques pueden acceder a los datos del bloque de datos globales. En cambio, el bloque de datos de instancia está asignado a un bloque de función, por lo que sus datos sólo están disponibles en este bloque de función (v. apt. 5.5).

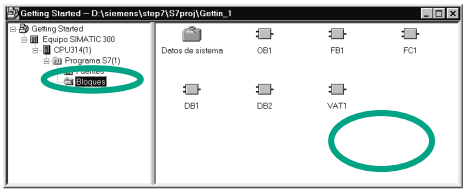
Antes de crear el bloque de datos globales tendría que familiarizarse con la programación en KOP, FUP o AWL (v. caps. 4 y 5), así como con la programación simbólica (v. cap. 3).



Si ha creado el proyecto de ejemplo "Getting Started" (caps. 1–7), ábralo.

Si no, cree ahora otro proyecto en el Administrador SIMATIC con **Archivo > Asistente "Nuevo proyecto"**. Proceda como hemos hecho en el apartado 2.1 y nombre el proyecto "Getting Started DB global".

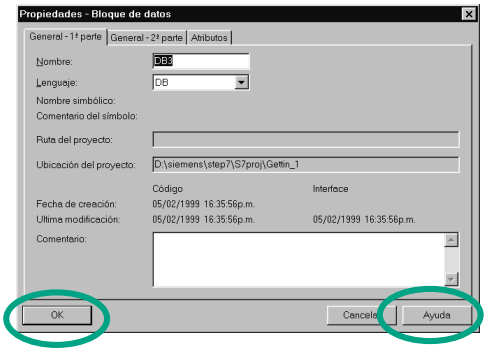
Nosotros nos basaremos en el proyecto "Getting Started". Sin embargo, puede seguir cada uno de los pasos igualmente con otro proyecto.



Navegue hasta la carpeta **Bloques** y ábrala.

Con la tecla derecha del ratón haga clic en la mitad derecha de la ventana.

Inserte un **bloque de datos** con ayuda del menú contextual.

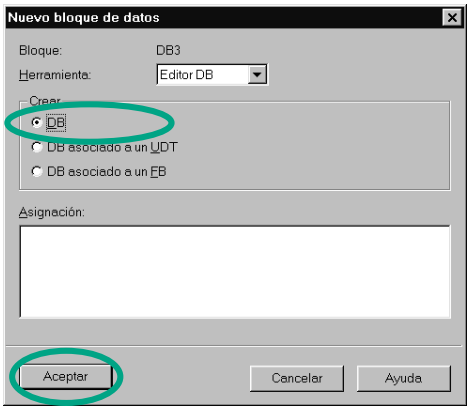


Confirme todos los ajustes estándar del cuadro de diálogo "Propiedades – Bloque de datos" con **Aceptar (OK)**.

Para más información consulte la **Ayuda**.

El bloque de datos DB3 se ha insertado en la carpeta **Bloques**.

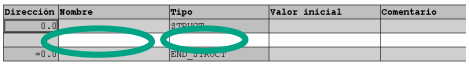
Abra el **DB3** haciendo un doble clic.



Active el **bloque de datos** en el cuadro de diálogo "Nuevo bloque de datos" que aparece a continuación. Seguidamente cierre el cuadro de diálogo con **Aceptar**.

Recuerde que en el apartado 5.5 hemos creado un bloque de datos de instancia activando la opción "DB asociado a un FB". Para crear un bloque de datos de instancia deberemos activar ahora la opción "DB".

Programar variables en el bloque de datos



En la columna 'Nombre' introduzca "Revol_reales_MG".

Para la columna 'Tipo' seleccione del menú contextual que aparece al pulsar la tecla derecha del ratón **Tipos de datos > Datos simples > INT**.

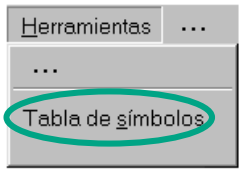
En el ejemplo hemos definido tres datos globales para el DB3. Introduzca los datos en la tabla de declaración de variables.

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	BM_Ist_Drehzahl	INT	0	Revoluciones reales del motor de gasolina
+2.0	DM_Ist_Drehzahl	INT	0	Revoluciones reales del motor diesel
+4.0	Soll_erreicht	BOOL	FALSE	Ambos motores han alcanzado el número teórico de revolucio
=6.0		END_STRUCT		

Las variables para los números reales de revoluciones "BM_Ist_Drehzahl" (Revol_reales MG) y "DM_Ist_Drehzahl" (Revol_reales_MD) del bloque de datos son tratadas del mismo modo que las palabras de marcas MW2 (BM_Ist_Drehzahl) y MW4 (MD_Ist_Drehzahl). Vea el capítulo siguiente.

Guarde el bloque de datos globales.

Asignar símbolos



A los bloques de datos también se les puede asignar nombres simbólicos.

Abra la **tabla de símbolos** e introduzca el símbolo "Datos_globales" para el bloque de datos DB3.

Si en el capítulo 4 ha copiado la tabla de símbolos de un proyecto de ejemplo (zEs01_02_STEP7__AWL_1-10, zEs01_06_STEP7__KOP_1-10 o zEs01_04_STEP7__FUP_1-10) en el proyecto "Getting Started", no tendrá que insertar más símbolos.

Símbolo	Dirección	Tipo de datos	Comentario
...	...	...	...
Datos_globales	DB 3	DB 3	Bloque de datos globales

Guarde la tabla de símbolos y cierre la ventana "Editor de símbolos".

Además, cierre la tabla de declaración de variables del bloque de datos globales.

Bloque de datos globales en la tabla de declaración de variables:

Con **Ver > Datos** puede modificar los valores actuales del tipo de datos INT en la tabla del bloque de datos globales (v. apt. 5.5).

Bloque de datos globales en la tabla de símbolos:

Contrariamente al bloque de datos de instancia, el tipo de datos del bloque de datos globales utilizado en la tabla de símbolos es la dirección absoluta. En nuestro ejemplo se trata del tipo de datos "DB3". En el bloque de datos de instancia se utiliza siempre el FB asignado.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques" y "Crear bloques de datos".

10 Programar una multiinstancia

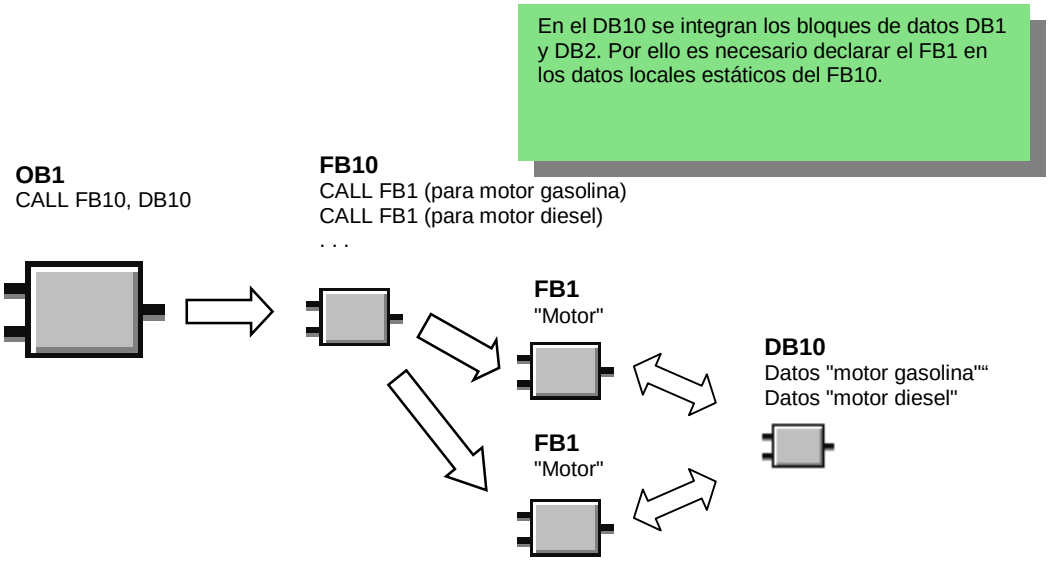
10.1 Crear y abrir un bloque de función de orden superior

En el capítulo 5 hemos programado un control de motor con el bloque de función "Motor" (FB1). En la llamada del bloque de función FB1 desde el OB1, el FB1 utilizaba sólo los bloques de datos "Gasolina" (DB1) y "Diesel" (DB2), los cuales contenían los datos de los correspondientes motores (p.ej., '#Drehzahl_Soll').

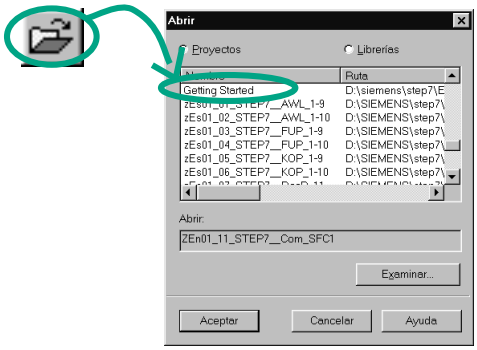
Supongamos que su tarea de automatización requiere otros controles de motor, p.ej., para un motor que funciona con aceite de colza, para un motor de hidrógeno, etc.

Procediendo de la misma forma que lo hemos hecho hasta ahora, utilizaríamos el FB1 asignándole un nuevo DB por cada control adicional. Dicho DB contendrá los datos del motor en cuestión. Así pues, para el control del motor de aceite de colza, utilizaríamos el FB1 con el DB3; para el motor de hidrógeno, el FB1 con el DB4; y así sucesivamente. Por consiguiente, el número de bloques aumentaría rápidamente al aumentar el número de controles programados.

Es posible reducir el número de bloques programando multiinstancias. Para ello se tiene que crear un nuevo FB de orden superior (en nuestro ejemplo, el FB10) y llamar desde el mismo al FB1, que ha permanecido invariable, como "instancia local". En cada llamada, el FB1 subordinado deposita sus datos en el bloque de datos DB10 del FB10 de orden superior, con lo cual no se tienen que asignar más DBs al FB1. Todos los FBs acceden a un único bloque de datos (aquí, el DB10).

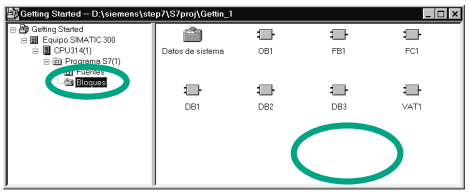


Antes de crear este bloque de función tendría que familiarizarse con la programación en KOP, FUP o AWL (v. caps. 4 y 5), así como con la programación simbólica (v. cap. 3).



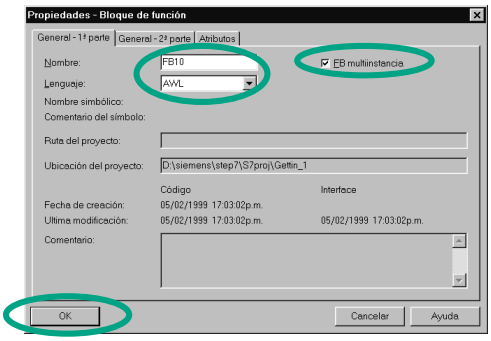
Si ha creado el ejemplo "Getting Started" (caps. 1–7), ábralo.

Si no, abra el proyecto zEs01_05_STEP7__KOP_1-9, zEs01_01_STEP7__AWL_1-9 o zEs01_03_STEP7__FUP_1-9 en el Administrador SIMATIC.



Navegue hasta la carpeta **Bloques** y ábrala.

Con la tecla derecha del ratón haga clic en la mitad derecha de la ventana y, con ayuda del menú contextual, inserte un bloque de función.



Cambie el nombre del bloque escribiendo "FB10" y elija el lenguaje deseado.

Dado el caso, active la opción **FB multiinstancia** y confirme los demás ajustes con **Aceptar**.

El FB10 se ha insertado en la carpeta 'Bloques'. Abra el **FB10** haciendo un doble clic.

Las multiinstancias se pueden crear para cualquier bloque de función, p.ej., para controlar válvulas. Si desea utilizar multiinstancias, no olvide que tanto el bloque de función invocante como los bloques de función llamados tienen que tener activada la opción "FB multiinstancia".

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques" y "Crear bloques y librerías".

10.2 Programar el FB10

Para llamar el FB1 como "instancia local“ del FB10, en la tabla de declaración de variables se declara para cada llamada del FB1 una variable estática indicándose en cada caso un nombre distinto. El tipo de datos utilizado es el FB1 ("Motor").

Rellenar la tabla de declaración de variables

La ventana KOP/AWL/FUP del programa está abierta. Declare las siguientes variables para la llamada del FB1.

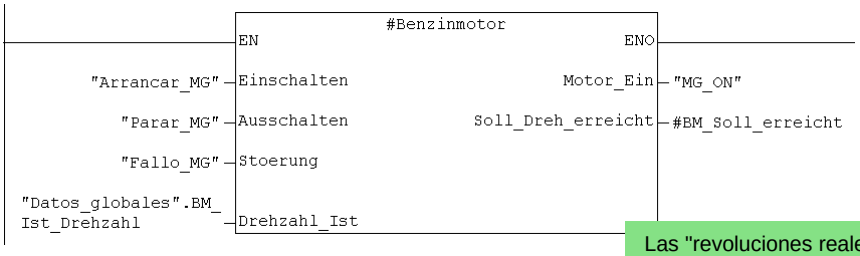
Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
	in				
0.0	out	Soll_erreicht	BOOL	FALSE	Ambos motores han alcanzado el número teórico de revoluciones
	in_out				
2.0	stat	Benzinmotor	"Motor"		Primera instancia local del FB 1 "Motor"
10.0	stat	Dieselmotor	"Motor"		Segunda instancia local del FB 1 "Motor"
0.0	temp	BM_Soll_erreicht	BOOL		Se ha alcanzado el número de revoluciones teórico (motor de gasc
0.1	temp	DM_Soll_erreicht	BOOL		Número de revoluciones teórico alcanzado (motor diesel)

Las instancias locales declaradas aparecerán en el catálogo de elementos del programa bajo "Multiinstancias".

Programar el FB10 en KOP

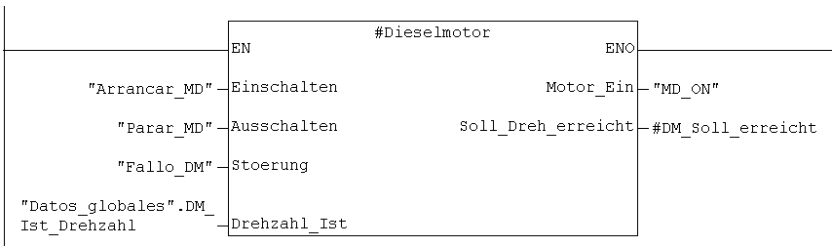
Inserte la llamada "Motor de gasolina" como bloque multiinstancia "Motor de gasolina" en el segmento 1.

A continuación inserte los contactos normalmente abiertos y complete la llamada con los nombres simbólicos.

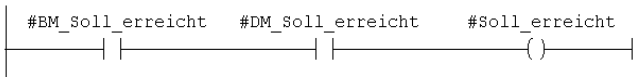


Las "revoluciones reales" de los motores no se recogen de ninguna marca (v. apt. 5.6 y siguientes), sino de un bloque de datos globales (v. apt. 9.1). El direccionamiento general es "Bloque de datos".Operando, p.ej., "Bloq_datos_glob".N°_real_rev_MG.

Inserte otro segmento y programe la llamada del motor diesel. Proceda como hemos hecho con el segmento 1.



Inserte otro segmento y programe una conexión en serie con la dirección correspondiente. A continuación guarde el programa y cierre el bloque.



Las variables temporales ("Alcanz_n°_real_rev_MG" y "Alcanz_n°_real_rev_MD“) se transfieren al parámetro de salida "Alcanz_n°_teór_rev", el cual se procesa luego en el OB1.

Programar el FB10 en AWL

```
CALL  #Benzinmotor
 Einschalten       :="Arrancar_MG"
 Ausschalten       :="Parar_MG"
 Stoerung          :="Fallo_MG"
 Drehzahl_Ist      :="Datos_globales".BM_Ist_Drehzahl
 Motor_Ein         :="MG_ON"
 Soll_Dreh_erreicht:=#BM_Soll_erreicht


CALL  #Dieselmotor
 Einschalten       :="Arrancar_MD"
 Ausschalten       :="Parar_MD"
 Stoerung          :="Fallo_DM"
 Drehzahl_Ist      :="Datos_globales".DM_Ist_Drehzahl
 Motor_Ein         :="MD_ON"
 Soll_Dreh_erreicht:=#DM_Soll_erreicht


U     #BM_Soll_erreicht
U     #DM_Soll_erreicht
=     #Soll_erreicht
```

Si programa en AWL, seleccione el campo de entrada que se encuentra debajo del nuevo segmento e introduzca las instrucciones AWL que le mostramos a la izquierda.

A continuación guarde el programa y cierre el bloque.

Programar el FB10 en FUP

Si programa en FUP, seleccione el campo de entrada que se encuentra debajo del nuevo segmento e introduzca las instrucciones FUP que le mostramos a continuación.

Guarde el programa y cierre el bloque.

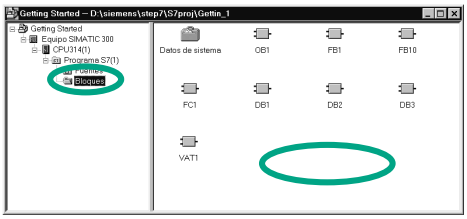
Para poder procesar las dos llamadas del FB1 en el FB10, tiene que llamarse primero al FB10.

Las multiinstancias sólo se pueden programar para bloques de función, y no para funciones (FCs).

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar bloques", "Crear bloques lógicos" y "Multiinstancias en la tabla de declaración de variables".

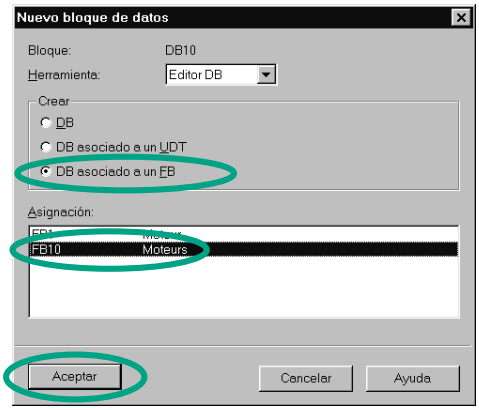
10.3 Crear el DB10 y adaptar el valor actual

El nuevo bloque de datos DB10 sustituye los bloques de datos DB1 y DB2. En el DB10 se depositan los datos de los motores de gasolina y diesel que se requerirán más tarde al llamar al FB10 en el OB1 (v. llamada del FB1 en el OB1, apts. 5.6 y siguientes).



En el Administrador SIMATIC, carpeta **Bloques** del proyecto "Getting Started", cree el bloque de datos DB10 con ayuda del menú contextual que aparece al pulsar la tecla derecha del ratón.

Cambie el nombre del bloque de datos escribiendo 'DB10' en el cuadro de diálogo y confirme los demás ajustes con **Aceptar**.



Se insertará el bloque de datos DB10. Abra el DB10 y se abrirá el cuadro de diálogo "Nuevo bloque de datos".

Active la opción **DB asociado a un FB** y seleccione el FB10.

Confirme los ajustes realizados con **Aceptar**.

Se abrirá el bloque de datos DB10. Active el comando de menú **Datos**.

Activando la vista 'Datos' aparecen todas las variables del DB10, incluyendo las variables "internas" de las dos llamadas del FB1 ("instancias locales").

En la vista 'Declaración' se visualizan las variables tal y como las hemos declarado en el FB10.

Cambie el valor actual del motor diesel poniéndolo a "1300", guarde el bloque de datos y ciérrelo.

Dirección	Declaración	Nombre	Tipo	Valor inicial	Valor actual	Comentario
0.0	out	Soll_erreicht	BOOL	FALSE	FALSE	Ambos motores han alcanzado el número teóri
2.0	stat:in	Benzinmotor.Eins	BOOL	FALSE	FALSE	Conectar motor
2.1	stat:in	Benzinmotor.Auss	BOOL	FALSE	FALSE	Desconectar motor
2.2	stat:in	Benzinmotor.Stoe	BOOL	FALSE	FALSE	Fallo del motor; conduce a una desconexión
4.0	stat:in	Benzinmotor.Dreh	INT	0	0	revoluciones reales del motor
6.0	stat:out	Benzinmotor.Motc	BOOL	FALSE	FALSE	Conexión del motor
6.1	stat:out	Benzinmotor.Soll	BOOL	FALSE	FALSE	Número de revoluciones alcanzado
8.0	stat	Benzinmotor.Dreh	INT	1500	1500	Número de revoluciones solicitado
10.0	stat:in	Dieselmotor.Eins	BOOL	FALSE	FALSE	Conectar motor
10.1	stat:in	Dieselmotor.Auss	BOOL	FALSE	FALSE	Desconectar motor
10.2	stat:in	Dieselmotor.Stoe	BOOL	FALSE	FALSE	Fallo del motor; conduce a una desconexión
12.0	stat:in	Dieselmotor.Dreh	INT	0	0	revoluciones reales del motor
14.0	stat:out	Dieselmotor.Motc	BOOL	FALSE	FALSE	Conexión del motor
14.1	stat:out	Dieselmotor.Soll	BOOL	FALSE	FALSE	Número de revoluciones alcanzado
16.0	stat	Dieselmotor.Dreh	INT	1500	1300	Número de revoluciones solicitado

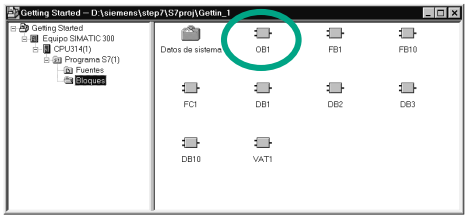
La tabla de declaración de variables del DB10 contiene ahora todas las variables. En la parte superior de la tabla se visualizan las variables utilizadas al llamar el bloque de función "Motor de gasolina" y, en la parte inferior, las utilizadas al llamar el bloque de función "Motor diesel" (v. apt. 5.5).

Las variables "internas" del FB1 mantienen sus nombres simbólicos, p.ej.: "Arrancar", que van precedidos del nombre de la instancia local, p.ej.: "Motor_gasolina.Arrancar“.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Programar lboques" y"Crear bloques de datos".

10.4 Llamada del FB10 en el OB1

La llamada del FB10 se realiza, en nuestro ejemplo, en el OB1. Esta llamada funciona de forma parecida a la llamada del FB1 desde el OB1 (v. apts. 5.6 y siguientes). Utilizando la multiinstancia se sustituirán los segmentos 4 y 5 programados en los apts. 5.6 y siguientes.



Abra el **OB1** en el proyecto en el que acaba de programar el FB10.

Si en el capítulo 4 ha copiado la tabla de símbolos de un proyecto de ejemplo (zEs01_06_STEP7__KOP_1-10, zEs01_02_STEP7__AWL_1-10 o zEs01_04_STEP7__FUP_1-10) en el proyecto "Getting Started", no tendrá que definir más símbolos.

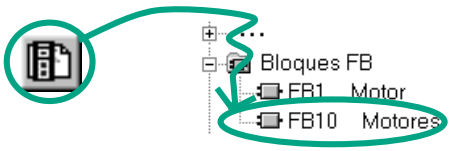
Definir nombres simbólicos

La ventana KOP/AWL/FUP está abierta. Abra la tabla de símbolos con **Herramientas > Tabla de símbolos** e introduzca en la tabla los nombres simbólicos para el bloque de función FB10 y para el bloque de datos DB10.

A continuación guarde la tabla de símbolos y cierre la ventana.

Símbolo	Dirección	po de datc	Comentario
...	...	...	...
Motores	FB 10	FB 10	Ejemplo de multiinstancias
Motores_datos	DB 10	FB 10	Bloque de datos de instancia para FB
...	...	...	...

Programar la llamada en KOP

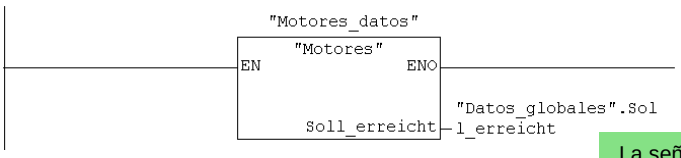


Inserte otro segmento al final del OB1 y añada con la llamada del **FB10** ("Motores").

Complete la llamada que le mostramos a continuación con los correspondientes nombres simbólicos.

Borre la llamada del FB1 en el OB1 (segmentos 4 y 5 de los apts. 5.6 y siguientes), pues el FB1 será llamado de forma central desde el FB10.

A continuación guarde el programa y cierre el bloque.



La señal de salida "Debe_revol_alcanzado" del FB10 ("Motores") se transfiere a la variable del bloque de datos globales.

Programar la llamada en AWL

Si programa en AWL, seleccione el campo de entrada que se encuentra debajo del nuevo segmento e introduzca las instrucciones AWL que le mostramos a continuación. Para ello utilice **Bloques FB > Motores FB10** del catálogo de elementos del programa.

Borre la llamada del FB1 en el OB1 (segmentos 4 y 5 de los apts. 5.6 y siguientes), pues el FB1 será llamado de forma central desde el FB10.

A continuación guarde el programa y cierre el bloque.

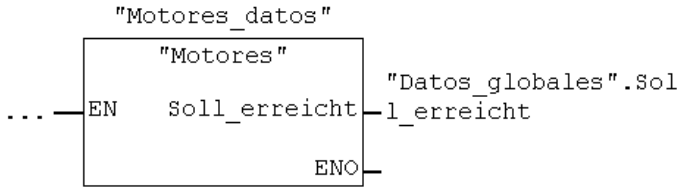
```
CALL  "Motores" , "Motores_datos"
 Soll_erreicht:="Datos_globales".Soll_erreicht
```

Programar la llamada en FUP

Si programa en FUP, seleccione el campo de entrada que se encuentra debajo del nuevo segmento e introduzca las instrucciones FUP que le mostramos a continuación. Para ello utilice **Bloques FB > Motores FB10** del catálogo de elementos del programa.

Borre la llamada del FB1 en el OB1 (segmentos 4 y 5 de los apts. 5.6 y siguientes), pues el FB1 será llamado de forma central desde el FB10.

A continuación guarde el programa y cierre el bloque.



Si su tarea de automatización controla otros motores, p.ej., motores que funcionen con gas natural, con biogás, etc., estos motores deberán programarse como multiinstancias, debiendo ser llamadas desde el FB10.

Para ello declare estos motores en la tabla de declaración de variables del FB10 ("Motores") como se indica en la figura y programe la llamada del FB1 (multiinstancia en el catálogo de los elementos del programa) en el FB10. A continuación deberá llevar a cabo la programación simbólica, es decir, definir los nuevos nombres simbólicos en la tabla de símbolos, p.ej., para la conexión/desconexión.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Cómo acceder a las ayudas de refeencia", "Saltos en descripciones del lenguaje KOP/FUP o AWL" y "Control de programa" o "Operaciones de control del programa".

11 Configurar la periferia descentralizada

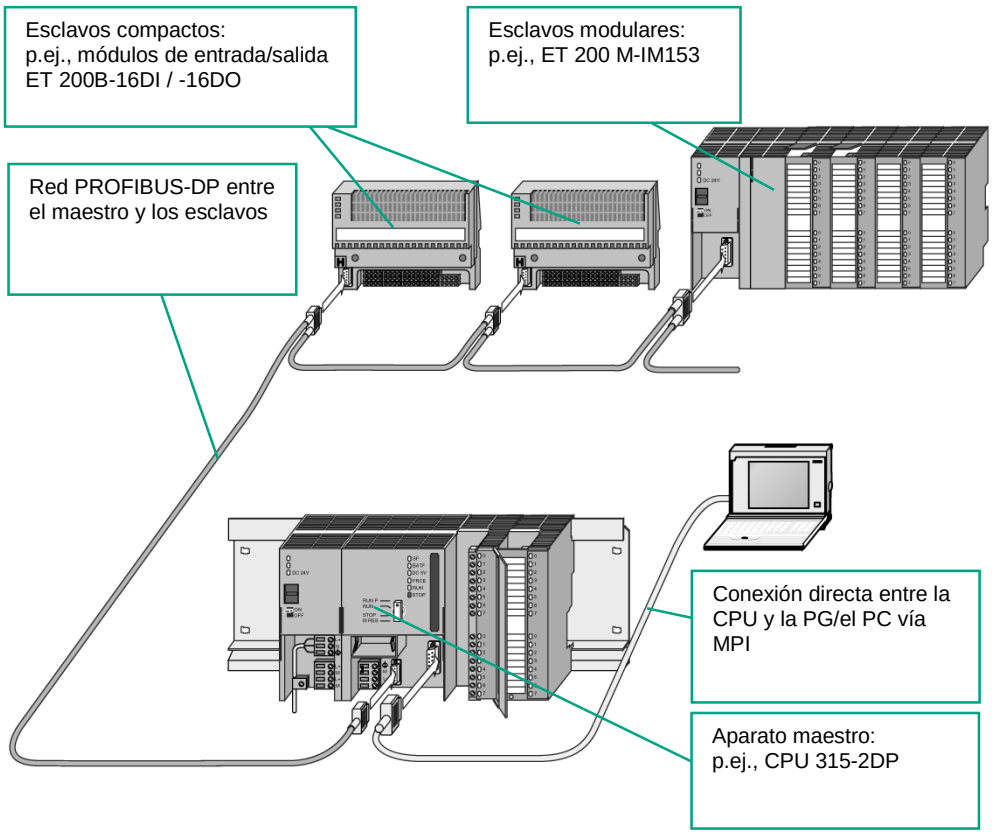
11.1 Conectar la periferia descentralizada vía el bus de campo PROFIBUS-DP

En las instalaciones que se han automatizado de forma convencional las conexiones por cable de los sensores y actuadores se enchufan directamente en los módulos de entrada/salida del sistema de automatización central, lo que comporta a menudo un cableado muy complejo.

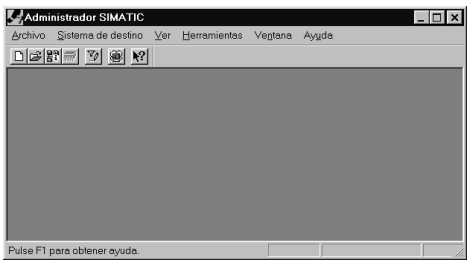
Las configuraciones descentralizadas minimizan considerablemente el cableado permitiendo posicionar los módulos de entrada/salida cerca de los sensores y actuadores. La conexión entre el sistema de automatización, los módulos periféricos y los aparatos de campo se establece vía el bus de campo PROFIBUS-DP.

En el capítulo 6 hemos aprendido a programar de forma convencional. Las configuraciones descentralizadas no se distinguen apenas de las configuraciones centralizadas. Basta con seleccionar del catálogo de hardware los módulos que se vayan a utilizar, disponerlos en el bastidor y adaptar sus propiedades a las exigencias del sistema en cuestión.

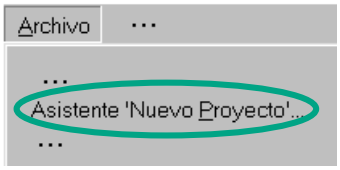
Para ello conviene saber cómo crear un proyecto y cómo realizar una configuración centralizada (v. apt. 2.1 y cap. 6).



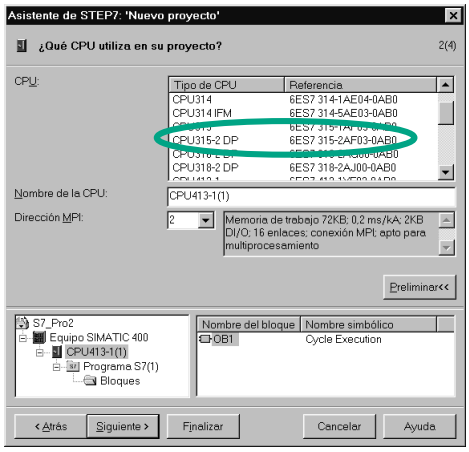
Crear un nuevo proyecto



Para crear un nuevo proyecto partiremos del Administrador SIMATIC. Cierre los demás proyectos que tenga abiertos para evitar confusiones.



Cree un **nuevo proyecto**.

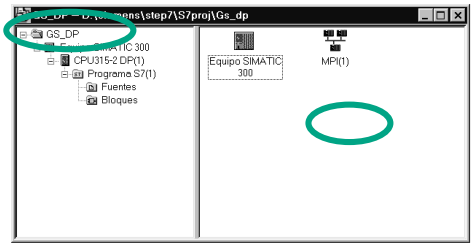


Seleccione la **CPU 315-2DP** (CPU con red PROFIBUS-DP) en el cuadro de diálogo que le mostramos a la izquierda.

Por lo demás proceda como hemos hecho en el apartado 2.1 y dé el nombre "GS-DP" (Getting Started – Periferia descentralizada) al proyecto.

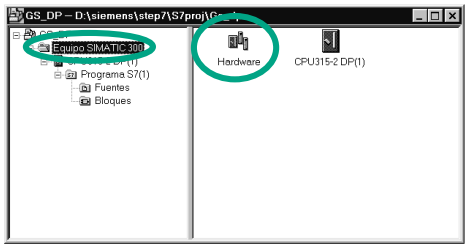
Si desea empezar con la configuración ahora mismo, indique la CPU que desea utilizar. Asegúrese de que es apta para periferia descentralizada.

Insertar la red PROFIBUS

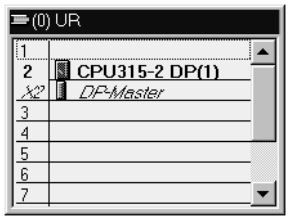


Seleccione la carpeta **GS-DP** e inserte la **red PROFIBUS** con ayuda del menú contextual que aparece al pulsar la tecla derecha del ratón.

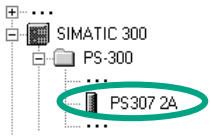
Configurar el equipo



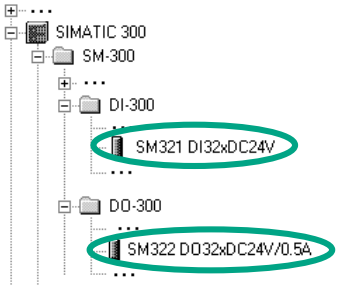
Seleccione la carpeta **Equipo SIMATIC 300** y haga doble clic en **Hardware**.
Se abrirá la ventana "HW Config" (v. apt. 6.1).



La CPU 315-2DP ya se encuentra en el bastidor. Si es necesario, active el catálogo de hardware eligiendo **Ver > Catálogo de hardware** o pulsando el botón correspondiente de la barra de herramientas.



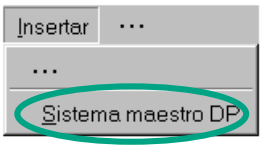
Inserte la fuente de alimentación **PS307 2A** en el slot 1 con 'Drag and Drop' (arrastrar y soltar).



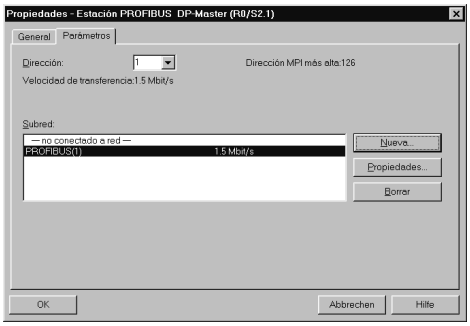
Inserte en los slots 4 y 5 los módulos de entradas/salidas **DI32xDC24V** y **DO32xDC24V/0.5A**.

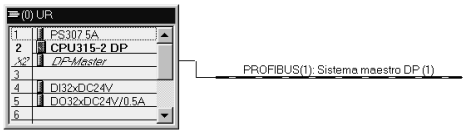
Además de la CPU apta para periferia descentralizada, puede posicionar otros módulos centrales en el mismo bastidor (nuestro ejemplo no recoge esta posibilidad).

Configurar el sistema maestro DP

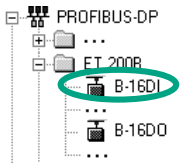


Seleccione el maestro DP del slot 2.1 e inserte un **Sistema maestro DP**.

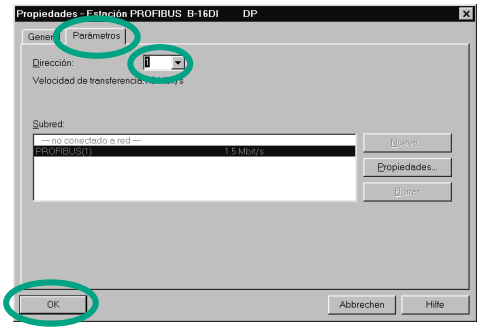




Todos los objetos del sistema maestro se pueden desplazar seleccionándolos y arrastrándolos hasta la posición deseada manteniendo pulsada la tecla del ratón.

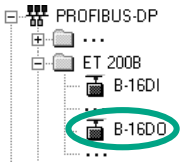


Navegue por el catálogo de hardware hasta el módulo **B-16DI** e insérte lo en el sistema maestro (arrastrándolo directamente hasta el sistema maestro con 'Drag and Drop' y soltándolo cuando el puntero del ratón cambie de forma).



En la ficha **Parámetros** del cuadro de diálogo "Propiedades" se puede modificar la dirección de estación del módulo recién insertado.

Confirme la dirección propuesta por el sistema, la **1**, con **Aceptar**.

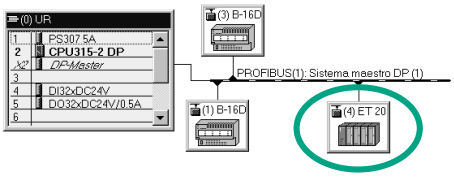


Acople del mismo modo el módulo **B-16DO** al sistema maestro.

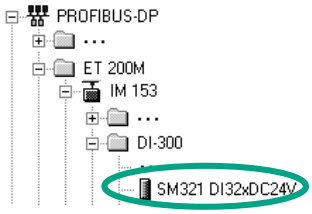
La dirección de estación se ajustará de forma automática en el cuadro de diálogo. Confírmela con **Aceptar**.

Arrastre el módulo interfase **IM153** hasta el sistema maestro y confirme la dirección de estación con **Aceptar**.

En nuestro ejemplo adoptamos las direcciones de estación estándar. Sin embargo, se pueden modificar adaptándolas a las necesidades de cada uno.



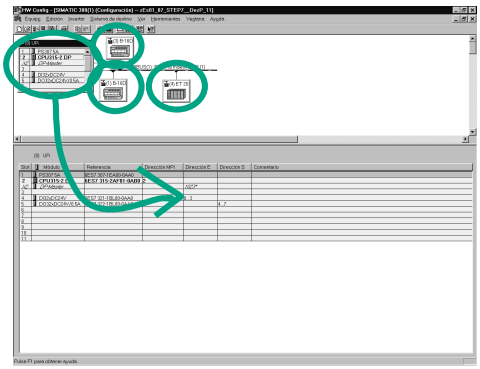
Seleccione el **ET200M** en la red.
En la tabla de configuración inferior se marcan los slots libres del ET200M.
Seleccione el sot **4**.



El ET200M puede acoger otros módulos de entrada/salida. P.ej.: elija el módulo **DI32xDC24V** para el slot 4 e insértelo con un doble clic.

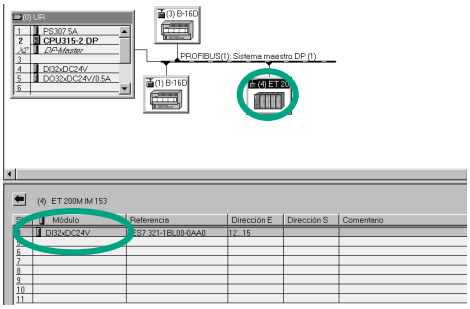
Al seleccionar módulos asegúrese de que se encuentra en la carpeta correcta del catálogo de hardware. P.ej.: si desea seleccionar módulos para el ET200M, asegúrese de que se encuentra en la carpeta 'ET200M'.

Modificar la dirección de estación



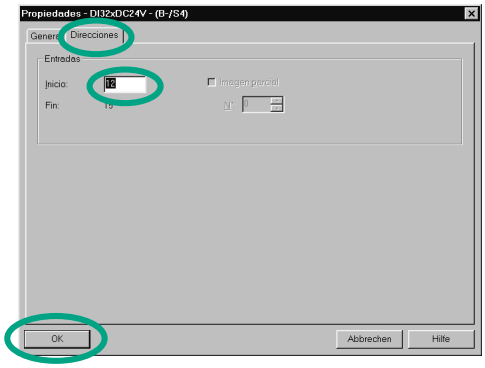
En nuestro ejemplo no es necesario modificar la dirección de estación. Sin embargo, en la práctica es lo más habitual.

Seleccione sucesivamente las demás estaciones y compruebe las direcciones de entrada y salida. Al configurar el hardware se han adaptado todas las direcciones, por lo que no encontrará ninguna que se haya asignado dos veces.

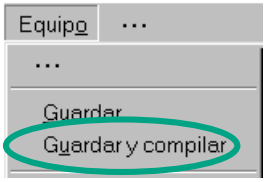


Supongamos que desea modificar la dirección del ET200M:

Seleccione **ET200M** y haga doble clic en **DO32xDC24V/0.4A** (slot 4).



Cambie la dirección 6 por la **12** en la ficha **Direcciones** del cuadro de diálogo "Propiedades".
Cierre el cuadro de diálogo con **Aceptar**.



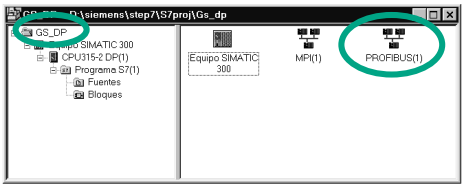
Finalmente active el comando **Guardar y compilar** para la configuración de la periferia descentralizada.

Cierre la ventana.

Activando 'Guardar y compilar' se comprobará automáticamente si la configuración es coherente. Si lo es, es decir, si no contiene errores, se generarán los datos del sistema, pudiéndose cargar entonces en el sistema de destino.

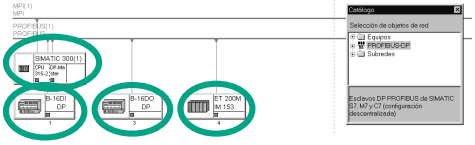
Si se activa 'Guardar', la configuración se guardará aunque contenga errores, pero no se podrá cargar en el sistema de destino.

Opcional: configuración de la red



La periferia descentralizada también se puede configurar al mismo tiempo que la red.

Para ello haga doble clic en la red **PROFIBUS (1)** desde el Administrador SIMATIC.



Entonces se abrirá la ventana "NETPRO“.

Si lo desea, puede acoplar más esclavos DP en el PROFIBUS-DP arrastrándolos desde el catálogo de objetos de la red.

Para configurar un elemento, haga doble clic en el mismo. Entonces se abrirá la ventana "Configurar hardware".

Con los comandos **Equipo > Comprobar coherencia** (ventana 'Configurar hardware') y **Red > Comprobar coherencia** (ventana 'NETPRO') se puede comprobar, antes de guardar una configuración, si ésta contiene errores. En caso afirmativo, éstos se visualizarán en STEP 7. Paralelamente se mostrarán posibles soluciones.

Para más información elija **Ayuda > Temas de Ayuda** y vaya a los temas "Configurar el hardware" y "Configurar la periferia descentralizada".

¡Felicidades! Acaba de terminar el proyecto de ejemplo "Getting Started". Esperemos que le haya servido para tener una visión general de STEP 7, para conocer los conceptos básicos, los procedimientos más habituales y las funciones más importantes de este paquete de software. Ahora puede pasar a la práctica creando su primer proyecto.

Si le interesan determinadas funciones de STEP 7 o se ha olvidado de las que aprendió en su día, le recomendamos consulte la Ayuda en pantalla de STEP 7.

Si desea profundizar en sus conocimimentos de STEP 7, no dude en inscribirse en uno de nuestros cursillos especiales. Diríjase al departamento de ventas y le informarán más detalladamente.

¡Le deseamos mucho éxito en sus proyectos futuros!

Siemens AG

A. Anexo A

A.1 Proyectos de ejemplo del Getting Started

- **zEs01_01_STEP7__AWL_1-9:**
  Caps. 1-9 programados en el lenguaje AWL, tabla de símbolos incluida.
- **zEs01_02_STEP7__AWL_1-10:**
  Caps. 1-10 programados en el lenguaje AWL, tabla de símbolos incluida.
- **zEs01_03_STEP7__FUP_1-9:**
  Caps. 1-9 programados en el lenguaje FUP, tabla de símbolos incluida.
- **zEs01_04_STEP7__FUP_1-10:**
  Caps. 1-10 programados en el lenguaje FUP, tabla de símbolos incluida.
- **zEs01_05_STEP7__KOP_1-9:**
  Caps. 1-9 programados en el lenguaje KOP, tabla de símbolos incluida.
- **zEs01_06_STEP7__KOP_1-10:**
  Caps. 1-10 programados en el lenguaje KOP, tabla de símbolos incluida.
- **zEs01_07_STEP7__DezP_11:**
  Cap. 11 programado con la periferia descentralizada.

Indice alfabético

OB1 4-2
Abrir el OB1 *4-2*
Abrir la función 8-1
Abrir un bloque de función 5-1
Administrador SIMATIC
  estructura del proyecto 2-4
Aplicar tensión 7-3
Arrancar el Administrador SIMATIC 2-1
AWL
  instrucción de memorización 4-10
  instrucción O 4-9
  instrucción U (Y) *4-8*
  llamada del bloque 5-16
  probar el programa 7-6
  programar la función de temporización 8-4
Ayuda
  llamar ***2-5***
Bloque de datos
  crear el DB de instancia 5-11
Bloque de datos globales
  crear 9-1
  programar 9-1
Bloque de datos globales en la tabla de declaración de variables 9-4
Bloque de datos globales en la tabla de símbolos 9-4
Bloque de función
  abrir 5-1
  crear 5-1
Borrar totalmente la CPU y ponerla a RUN 7-3
Búfer de diagnóstico
  evaluar 7-12
Cargar bloques de uno en uno 7-5
Cargar el programa en el sistema de destino 7-3
Comprobar coherencia
  equipo 11-7
  red 11-7
Conectar la CPU 7-5
Conectar la tabla de variables online 7-9
Conexión en paralelo en KOP 4-6
Conexión en serie en KOP 4-4
Conexión online
  configurar 7-1
Configuración de la red 11-7
Configurar el hardware 6-1, 7-1
Configurar el PROFIBUS-DP 11-1
Configurar el sistema maestro DP 11-4
Configurar la periferia descentralizada 11-1
Configurar los módulos centrales 6-1
Consultar la información del módulo 7-12
Copiar la tabla de símbolos *4-2*
Crear bloques de datos de instancia 5-11
Crear el bloque de datos globales 9-1
Crear el proyecto 2-1
Crear la función 8-1
Crear la tabla de variables 7-8
Crear un bloque de función 5-1
Datos 10-6
Declaración 10-6
Dirección de estación
  modificar 11-6
Direcciones absolutas 3-1
Editor de símbolos 3-2
Elegir uno de los lenguajes KOP, AWL o FUP *4-1*
Equipo > Comprobar coherencia 11-7
Establecer una conexión online 7-1
Estructura del proyecto en el Administrador SIMATIC 2-4
Evaluar el búfer de diagnóstico 7-12
FB
  programar en AWL 5-6
  programar en FUP 5-8
  programar en KOP 5-3
Forzar variables 7-10
Función
  abrir 8-1
  crear 8-1
  llamada 8-6
Función de memorización en FUP 4-14
Función de memorización en KOP 4-7
Función O (OR) 1-1
Función O en FUP 4-13
Función SR 1-2
Función U (Y) en FUP 4-11
Función Y (AND) 1-1
FUP
  función de memorización 4-14
  función O 4-13
  función U (Y) 4-11
  llamada del bloque 5-18
  probar el programa 7-6
  programar la función de temporización 8-5
  Hardware configurar 6-1
Insertar > Símbolo 4-5, 4-9, 4-12
Insertar símbolo

KOP 4-5
Insertar un símbolo
AWL 4-9
FUP 4-12
Instalación 1-5
Instrucción de memorización en AWL 4-10
Instrucción O en AWL 4-9
Instrucción U (Y) en AWL 4-8
Introducción a STEP 7 1-1
KOP
conexión en paralelo *4-6*
conexión en serie *4-4*
función de memorización *4-7*
llamada del bloque 5-13
probar el programa 7-6
programar la función de temporización 8-3
Llamada de la función 8-6
Llamada del bloque en AWL 5-16
Llamada del bloque en FUP 5-18
Llamada del bloque en KOP 5-13
Llamar la Ayuda 2-5
Modificar direcciones de estación 11-6
Modificar valores actuales 5-11
Multiinstancia
programar 10-1
Navegar en la estructura del proyecto ***2-5***
Observar variables 7-10
Periferia descentralizada
configurar 11-1
Planteamiento de la tarea 1-1
Probar el programa con AWL 7-6
Probar el programa con FUP 7-6
Probar el programa con KOP 7-6
Procedimiento a seguir con STEP 7 1-4
Programa
cargar el programa en el sistema de destino 7-3
Programación simbólica 3-2
Programar con FBs y DBs 5-1
Programar el FB en KOP 5-3
Programar el FB1 en AWL 5-6
Programar el FB1 en FUP 5-8
Programar el OB1 *4-1*
Programar la función de temporización en AWL 8-4
Programar la función de temporización en FUP 8-5
Programar la función de temporización en KOP 8-3
Programar online 7-5
Programar un bloque de datos globales 9-1
Programar una función (FC) 8-1
Programar una multiinstancia 10-1
Proyecto
crear 2-1
Proyectos de ejemplo A-1
Red > Comprobar coherencia 11-7
Rellenar la tabla de declaración de variables
AWL 5-6
FUP 5-8
Representación simbólica
AWL 4-10
FUP 4-14
KOP 4-7
Simbólica
programación 3-2
Sistema maestro DP
configurar 11-4
Software de ampliación SIMATIC 2-6
STEP7
Asistente
Nuevo proyecto 2-1
Tabla de símbolos 3-2
copiar 4-2
Tabla de variables
conectar online 7-9
crear 7-8
Tipo de datos 3-3
Variables
forzar 7-10
observar 7-10
Ventana KOP/AWL/FUP *4-3*

Siemens AG
A&D AS E 81

Oestliche Rheinbrueckenstr. 50
D-76181 Karlsruhe
R.F.A.

Remitente:

Nombre: _
Cargo: _
Empresa: _
Calle: _
Código postal: _
Población: _
País: _
Teléfono: _

Indique el ramo de la industria al que pertenece:

- ❒ Industria del automóvil
- ❒ Industria química
- ❒ Industria eléctrica
- ❒ Industria alimentaria
- ❒ Control e instrumentación
- ❒ Industria mecánica
- ❒ Industria petroquímica
- ❒ Industria farmacéutica
- ❒ Industria del plástico
- ❒ Industria papelera
- ❒ Industria textil
- ❒ Transportes
- ❒ Otros _ _ _ _ _ _ _ _ _ _ _

Observaciones/sugerencias

Sus observaciones y sugerencias nos permiten mejorar la calidad y utilidad de nuestra documentación. Por ello le rogamos que rellene el presente formulario y lo envíe a Siemens.

Responda por favor a las siguientes preguntas dando una puntuación comprendida entre 1 = muy bien y 5 = muy mal

1. ¿ Corresponde el contenido del manual a sus exigencias ? ☐
2. ¿ Resulta fácil localizar las informaciones requeridas ? ☐
3. ¿ Es comprensible el texto ? ☐
4. ¿ Corresponde el nivel de los detalles técnicos a sus exigencias ? ☐
5. ¿ Qué opina de la calidad de las ilustraciones y tablas ? ☐

En las líneas siguientes puede exponer los problemas concretos que se le hayan planteado al manejar el manual:

_ _ _ _